UNIVERSITÉ D'AIX-MARSEILLE II - MÉDITERRANÉE
U.F.R. M.I.M.

THÈSE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ D'AIX-MARSEILLE II

Spécialité Informatique

par

Tristan COLOMBO

Algorithmes pour la recherche de classes de gènes en relations fonctionnelles par analyse de proximités et de similarités de séquences

---

Soutenue le 7 décembre 2004 devant le jury composé de :

M.	François Denis	LIF	–	Univ. Aix-Marseille I	Président
M.	Alain Guénoche	IML	–	CNRS	Directeur de thèse
M.	Yves Quentin	LCB	–	CNRS	Directeur de thèse
M.	Guy Perrière	LBBE	–	Univ. Lyon I	Rapporteur
M.	Thomas Schiex	BIA	–	INRA	Rapporteur

Laboratoire de Chimie Bactérienne
Institut de Mathématiques de Luminy

UNIVERSITÉ D'AIX-MARSEILLE II - MÉDITERRANÉE
U.F.R. M.I.M.

THÈSE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ D'AIX-MARSEILLE II

Spécialité Informatique

par

Tristan COLOMBO

Algorithmes pour la recherche de classes de gènes en relations fonctionnelles par analyse de proximités et de similarités de séquences

---

Soutenue le 7 décembre 2004 devant le jury composé de :

M.	François Denis	LIF	–	Univ. Aix-Marseille I	Président
M.	Alain Guénoche	IML	–	CNRS	Directeur de thèse
M.	Yves Quentin	LCB	–	CNRS	Directeur de thèse
M.	Guy Perrière	LBBE	–	Univ. Lyon I	Rapporteur
M.	Thomas Schiex	BIA	–	INRA	Rapporteur

Laboratoire de Chimie Bactérienne
Institut de Mathématiques de Luminy

A ma mère et à mon père,

A Clément,

A mes grands-parents : Odette, Carmélina, Marius,

A toute ma famille,

A tous mes amis.

Le monde juge bien des choses,
car il est dans l'ignorance naturelle
qui est le vrai siège de l'homme.
Les sciences ont deux extrémités qui se touchent,
la première est la pure ignorance naturelle
où se trouvent tous les hommes en naissant,
l'autre extrémité est celle où arrivent les grandes âmes qui,
ayant parcouru tout ce que les hommes peuvent savoir,
trouvent qu'ils ne savent rien et se rencontrent en cette même ignorance
d'où ils étaient partis, mais c'est une ignorance savante qui se connaît.
Pascal

Remerciements

Je voudrais, tout d'abord, remercier tous les membres du jury : Monsieur Guy Perrière et Monsieur Thomas Schiex qui m'ont fait l'honneur d'être les rapporteurs de ce mémoire, et Monsieur François Denis qui a accepté de participer à mon jury.

Je tiens à remercier Alain Guénoche et Yves Quentin qui ont accepté d'être mes directeurs de thèse. Merci à eux pour leur constante disponibilité et leur vision de la bioinformatique : ils ont su guider mon travail au travers des méandres de la biologie.

Je souhaite également remercier Marc Chippaux puis Frédéric Barras pour m'avoir accueilli dans leur laboratoire, François Denizot pour son aide et plus généralement tout le personnel du laboratoire pour leur accueil et pour l'ambiance agréable qui règne dans ces locaux.

Je remercierai aussi :

Philippe Notareschi, pour son soutien constant, ses encouragements, son aide précieuse et son enthousiasme pour la configuration de la Debian Linux, nos projets informatique inachevés qui ne le resteront pas, nos pauses sportives : roller, street-hockey, vélo, course à pied, beach volley ... et tout le reste. Merci.

Frédéric Nicolas pour cette année passée au laboratoire où nous avons pu travailler et progresser ensemble dans la compréhension du langage R. Cette année s'est écoulée bien vite ...

Philippe Jégou, mon tuteur de monitorat, pour ces quatre années passées à enseigner à l'Université des Sciences et Techniques de Saint-Jérôme et pour la confiance qu'il m'a toujours accordée dans ces enseignements.

Belaïd Benhamou pour nos discussion sur les problèmes de satisfaction de contraintes et son aide pour la résolution du problème d'exploration du voisinage d'un gène.

Cécile Capponi pour ses divers conseils et son encadrement lors du stage de Dea.

Gwennaele Fichant pour son aide dans la relecture des documents et les répétitions des diverses présentations.

Les divers professeurs que j'ai rencontré tout au long de mon apprentissage à l'université et dont les cours m'ont particulièrement marqué : Bruno Durand et Enrico Formenti qui par leur sujet de mémoire de maîtrise m'ont ouvert les portes de la bioinformatique, Amar Oussalah pour ses cours de langage C, Georges-André Silber pour ses TP de réseau, Laurence Pierre pour ses TD d'Architecture des ordinateurs et pour sa gentillesse et sa disponibilité continuelle pour les étudiants.

Séverine Bérard et Cyril Terrioux pour leur relecture de la thèse.

Henri Bellange, mon professeur de "logique cablée" en Deug. Si je suis arrivé jusque là c'est un peu à cause de lui ...

Stéphane Kazarian, mon "oncle", qui m'a initié aux joies du montage/démontage des ordinateurs et m'a dépanné plus d'une fois ... sans lui il m'aurait été difficile de commencer à programmer ...

Christian et Mireille Ozoux pour leur aide ô combien précieuse lors de la correction des documents en langue anglaise.

Julia Bruneau pour son soutien et nos looOoongues heures de discussion ...

Aline Sibenaler et Maryan Sidorkiewicz parce-que :
Je marchais au hasard le soir était tombé
Avec mon sac et ma raquette j'étais un peu fatigué
Tout était si désert, où me désaltérer ?
Et puis j'ai vu de la lumière et je vous ai trouvés.

Clément, mon petit frère, pour son aide en biologie, ses encouragements et son soutien sportif (même si normalement quand on soutient quelqu'un on évite de le lâcher dans les cols ...).

Enfin, je tiens à remercier tout particulièrement ma mère et mon père qui pendant ces quelques 24 années d'études m'ont toujours soutenu et appuyé dans mes choix. Ma mère qui m'a appris à lire et à écrire et qui par là même m'a ouvert les portes du savoir, et mon père qui m'a "poussé" sur les chemins de l'informatique et de l'électronique. Merci pour leur grande patience et tout, tout le reste ...
Merci à mon grand-père qui s'est toujours intéressé à mon travail et qui, pendant ces trois années, a voulu comprendre le sens de mes recherches, et plus généralement à toute ma famille, témoins de mes joies, de mes enthousiasmes et de mes déceptions.

Table des matières

Introduction

Le traitement informatique des séquences biologiques a fait son apparition à la fin des années 1970 avec la généralisation du séquençage et la création des premières banques de données de séquences d'ADN (EMBL et GenBank). Au fil des années, les techniques de séquençage s'améliorant grâce aux robots, aux ordinateurs et aux algorithmes développés, la détermination du génome complet de bactéries (Fleishmann *et col.*, 1995) puis d'organismes eucaryotes (Bussey *et col.*, 1997) est devenue possible. Ces progrès ont contribué à l'essor d'une nouvelle branche théorique de la biologie, la bioinformatique dont le but est d'effectuer la synthèse des données disponibles à l'aide de modèles et de théories, d'énoncer des hypothèses généralisatrices, et de formuler des prédictions à partir d'une approche par modélisation appliquée à des objets formalisés (Claverie *et col.*, 2000). Elle est interdisciplinaire par nature car à l'intersection de quatre disciplines scientifiques : la biologie, la physique, les mathématiques et l'informatique. On peut dégager deux thèmes majeurs de la bioinformatique moléculaire :

- La compilation et l'organisation des données : les bases (et banques) de données (thématiques ou non) constituent une source de connaissance d'une grande richesse que l'on peut exploiter dans le développement de méthodes d'analyse ou de prédiction.
- L'analyse de données des séquences : l'objectif principal est de repérer ou caractériser une fonctionnalité ou un élément biologique intéressant. On retrouve dans cette catégorie les problèmes bioinformatiques au centre de la "génomique fonctionnelle" (Rocha, 2000) :
  - L'annotation syntaxique : Une fois déterminée, la séquence génomique ne représente qu'une donnée brute qu'il faut déchiffrer (identification de zones codant potentiellement pour des protéines, de séquences promotrices, de phases codantes sur une molécule d'ADN, ...). L'annotation syntaxique de certains éléments, tels que les gènes codant pour des protéines de génomes procaryotes, ne pose désormais presque plus de problème. En revanche, l'identification de signaux de régulations reste un problème d'actualité.
  - L'annotation fonctionnelle : On attribue des fonctions biologiques aux données détectées lors de l'annotation syntaxique. Cette opération se fait soit grâce à des données expérimentales, soit par recherche de séquences fortement similaires et analogie – il s'agit alors de prédictions. Cette étape dépend énormément de la qualité des informations qui lui sont transmises et une erreur à ce niveau pourra très vite être propagée à d'autres données.
  - L'annotation relationnelle : Il s'agit d'identifier les relations existant entre les objets caractérisés lors des étapes d'annotation syntaxique et fonctionnelle : implication dans un même processus cellulaire, interaction physique de protéines ... Les données manipulées ici nécessitent un haut degré d'abstraction et de structuration ; elles sont généralement représentées sous forme de graphe.

L'aspect théorique de la bioinformatique, notamment en terme de prédiction, contribue à un gain de temps (et un gain financier) par rapport à ce que constituerait une expérimentation

"aveugle" (Hinton, 1997). Cela permet d'éviter de vérifier et de tester beaucoup d'hypothèses qui auraient demandé un trop grand effort expérimental (Rocha, 2000). Ainsi, la bioinformatique est une approche globale, capable d'enrichir le domaine fondamental de connaissances nouvelles et d'être à l'origine de concepts biologiques originaux.

L'approche bioinformatique que nous avons choisie est de partir d'un problème biologique et de le résoudre en utilisant des méthodes informatiques que nous adaptons et, si possible, améliorons. Nous gardons toujours à l'esprit dans notre démarche que ces algorithmes doivent être développés jusqu'à devenir des "outils" destinés à être utilisés de manière simple au travers d'interfaces graphiques adaptées. De plus, les méthodes employées doivent être suffisamment rapides pour traiter une masse de données importante et en constante augmentation. De part les questions posées, nous nous inscrivons dans le domaine de la "génomique fonctionnelle" : à partir des séquences complètes de génomes, nous étudions le fonctionnement et l'évolution des organismes.

Ayant effectué ma thèse au sein du **L**aboratoire de **C**himie **B**actérienne dans l'équipe **G**énomique des **S**ystèmes **I**ntégrés, j'ai utilisé les transporteurs ABC[1] bactériens comme modèle d'étude. Ce sont des systèmes intégrés impliqués dans les échanges de molécules entre la bactérie et son milieu. L'analyse bioinformatique du répertoire de ces systèmes comprend l'identification des partenaires, l'assemblage, la reconstruction des systèmes incomplets, la classification en sous-familles, et l'identification du substrat transporté. Mon travail de thèse porte sur la résolution des problèmes rencontrés dans les étapes d'assemblage, de classification et de prédiction fonctionnelle. Les méthodes développées font appel à des algorithmes trouvant leurs fondements dans divers domaines de l'informatique. Nous utilisons les nombreuses données maintenant disponibles de génomes bactériens entièrement séquencés[2] et, nous nous plaçons dans un contexte évolutif. L'utilisation de relations évolutives est très intéressante car, bien que les gènes ne soient pas conservés en séquences, nous pouvons retrouver des relations de parenté entre des espèces différentes. Suivant la nature de la relation évolutive observée, les gènes possèderont alors des propriétés particulières telles que la conservation de la fonction du gène ancestral ou une modification de la fonction par spécialisation par exemple. Les fonctions de protéines qui sont connues dans certains génomes permettent d'effectuer des prédictions fonctionnelles dans d'autres génomes (ceci est très important pour les bactéries qui ne peuvent pas être manipulées en laboratoire).

La première partie de ce manuscrit présente les connaissances biologiques nécessaires pour suivre le travail effectué. Il s'agit d'une présentation générale des bactéries et de leur génome, puis des relations évolutives et fonctionnelles qui peuvent être déduites de l'analyse comparative de ces génomes. Cette partie se termine par une description de notre modèle d'étude, les transporteurs ABC, et par une description des différents problèmes abordés au cours de cette thèse.

Les trois parties suivantes sont conçues suivant le même canevas. J'expose tout d'abord le

[1] Une description détaillée est donnée dans le chapitre 1.

[2] Rappelons que le premier génome complet n'a vu le jour qu'en 1995. Actuellement, la banque de données du NCBI en compte 186 et au cours de cette thèse nous en avons utilisé 95.

problème et les hypothèses biologiques qui sont à la base du travail, puis, j'effectue un rappel du domaine informatique employé. En effet, les problèmes étant de natures assez diverses, les méthodes de résolution le sont également et font appel à des domaines informatiques différents. Enfin, je présente d'autres méthodes concurrentes et les résultats obtenus au cours de deux étapes : validation de la méthode puis application aux données biologiques.

Le premier de ces chapitres porte sur l'exploration du voisinage chromosomique d'un gène. Ce champ d'investigation est potentiellement important car, il permet de prédire des relations fonctionnelles entre gènes et de reconstruire ainsi des réseaux complexes comme par exemple les voies métaboliques. Au niveau des transporteurs ABC, cette étude peut permettre de préciser le substrat transporté par un système grâce à la conservation dans le même voisinage de gènes impliqués soit dans le métabolisme de ce substrat soit dans la régulation transcriptionnelle en réponse à un stimulus. L'originalité de notre approche repose sur la prise en compte de la distance entre les gènes comme mesure de l'intensité de la relation fonctionnelle et cela sans tenir compte de l'orientation des gènes. Ce problème a été traité en utilisant une méthode de résolution issue des problèmes de satisfaction de contraintes et donc une approche logique.

Dans le deuxième chapitre, j'aborde un problème de classification des transporteurs ABC en vue de constituer des groupes impliqués dans le transport de la même classe de substrat. Des travaux préliminaires ont montré l'existence d'une structuration en sous-familles de protéines liées au type de substrat transporté ((Tomii et Kanehisa, 1998), (Dassa *et col.*, 1999), (Quentin *et col.*, 2002)). Néanmoins, elles reposent sur de petits jeux de données et elles ne sont pas basées sur des critères rigoureux. Pour chaque domaine, en représentant les relations de similitudes par un graphe et en affectant à chaque sommet une valeur de densité déterminée en fonction de son degré local d'implication dans la structure du graphe, la recherche des zones de forte densité permet de déterminer des classes plus précises.

Enfin, dans le dernier chapitre, je présente une méthode permettant de généraliser la reconstruction des systèmes fonctionnels. En effet, la reconstruction de ces systèmes est basée sur deux règles émises à partir d'observations expérimentales : les gènes codant pour les différents partenaires de ces systèmes sont généralement voisins sur le chromosome et ces partenaires appartiennent à des sous-familles compatibles. Cependant, il arrive que des gènes codant pour certains partenaires soient absents du regroupement (on parlera de *système partiel*) ou que tout ou partie des gènes soient dispersés sur le chromosome (on parlera de *système éclaté*). Notre méthode répond à ces situations en exploitant les relations évolutives existant entre systèmes appartenant à différentes espèces de bactéries. Ainsi, lorsque les gènes codant pour les différents partenaires ne sont pas tous voisins sur le chromosome, notre méthode, basée sur une analyse de graphe, permet la reconstruction des systèmes fonctionnels.

Ces trois méthodes, en complément de l'identification des partenaires et de l'assemblage, permettent une étude fonctionnelle des transporteurs ABC. Pour finir, j'exposerai les perspectives offertes par ce travail.

1

De la biologie

Rien n'a de sens en biologie, si ce n'est à la lumière de l'évolution.
Théodosius Dobzhansky

Je tenterai ici de présenter de manière simple les connaissances biologiques indispensables à la compréhension de mon travail. Je vais plus particulièrement m'attacher à un domaine de la biologie, relativement récent, ayant émergé dans les années 1990 : la génomique. Pour m'aider dans ma tâche je me suis bien sûr inspiré de nombreux ouvrages de vulgarisation. Le premier de ces recueils a été écrit par un personnage incontournable de cet exercice de style : Jacquard (1992) et Jacquard et Kahn (2001). Il décrypte pour nous les rouages complexes de la vie et, plus important encore, soulève les questions morales, éthiques et plus généralement philosophiques que ces connaissances induisent. Je citerai également Douarin (2000) et Rensberger (2000) pour leurs diverses définitions. Danchin (1998), outre la biologie, aborde également l'utilisation d'outils informatiques pour explorer, découvrir de nouvelles connaissances. Et enfin, un ouvrage purement technique (Etienne, 1999), destiné plus spécifiquement aux biologistes, qui explique de manière très précise les mécanismes moléculaires du vivant.

1.1 Présentation générale des bactéries

Le règne du vivant se décompose en deux catégories : les procaryotes – dont les cellules ne comportent pas de noyau – et les eucaryotes – qui possèdent un noyau. Travaillant au sein de l'équipe **G**énomique des **S**ystèmes **I**ntégrés au **L**aboratoire de **C**himie **B**actérienne, je me suis plus particulièrement intéressé aux génomes procaryotes, autrement dit aux bactéries.

1.1.1 Taxonomie bactérienne

De l'ordre du micromètre, les bactéries sont les plus petits organismes vivants[3]. Une bactérie est une cellule entourée d'une membrane et contenant tous les éléments nécessaires à sa propre reproduction. Les populations de bactéries peuvent s'adapter aux variations de leur environnement, ou adopter en quelques générations seulement de tout nouveaux caractères. Les cellules bactériennes primitives ont donné naissance à deux grands groupes[4] :

- les archaebactéries : ce sont des organismes capables de se développer dans des conditions extrêmes telles que les glaces antarctiques, l'intérieur d'un réacteur nucléaire actif, la cheminée d'un volcan, ... De par ce fait on les appelle également parfois bactéries extrêmophiles. Elles se décomposent en trois sous-familles :
  - les halophiles, bactéries vivant en milieux hypersalés,
  - les méthanogènes, produisant du méthane et vivant au fond des mers ou dans le tube digestif de certains animaux,
  - les thermoacidophiles, vivant dans des milieux à la fois acides et chauds.
- les bactéries : cette famille englobe tous les autres procaryotes. Elle peut être elle aussi répartie en sous-familles. Une des classifications proposée est composée de trois sous-familles :
  - les cyanobactéries, bactéries utilisant une photosynthèse de type oxygénique,
  - les Gram+, bactéries réagissant positivement au test de Gram[5],
  - les Gram–, bactéries réagissant négativement au test de Gram[6].

La taxonomie correspond à la classification des organismes vivants. Chez les procaryotes, les critères de classification sont si nombreux qu'il existe plusieurs taxonomies dans lesquelles les bactéries sont réparties en classes, ordres, familles, *etc.* On peut les distinguer en fonction de leur morphologie – bacilles, cocci, ou spirilles[7] –, les caractéristiques de leur paroi cellulaire – réaction positive ou négative au test de Gram (Gram+ et Gram-) –, ou encore leurs besoins en oxygène – aérobies strictes, anaérobies strictes, ou aérobies facultatives, *etc.*

Sur Terre, les bactéries sont ubiquitaires ; elles présentent une grande diversité[8], et sont très anciennes : on estime leur apparition à trois milliards et demi d'années. Ces organismes sont très étudiés à cause de leur pouvoir de nuisance (bactéries pathogènes), ou leur intérêt économique (fermentation).

Les procaryotes ont généralement besoin d'humidité pour se développer et se reproduire et tous ont besoin de nutriments[9] pour obtenir les éléments de base de la matière vivante ; éléments organiques comme minéraux : carbone, oxygène, azote et phosphore en particulier. Ces différentes

[3] Les virus sont plus petits que les bactéries mais sont incapables de se reproduire ou d'effectuer des synthèses seuls. Peut-on alors les considérer comme des êtres vivants ?

[4] Comme dans toute classification, il en existe de nombreuses autres basées sur des critères différents.

[5] Le test de Gram permet d'identifier les bactéries par rapport à l'épaisseur de leur paroi : la paroi Gram+ est plus épaisse que la paroi Gram–. Les bactéries à Gram+ se colorient en bleu violet au test de Gram.

[6] Les bactéries à Gram– ont une paroi cellulaire beaucoup plus mince et d'aspect laminé.

[7] On distingue ces trois grands groupes de bactéries bien que de nombreuses espèces soient très polymorphes, adoptant une forme ou une autre suivant les conditions dans lesquelles elles sont placées.

[8] Nous savons aujourd'hui qu'entre 99% et 99,9% des bactéries nous sont inconnues (Whitman *et col.*, 1998).

[9] Ces nutriments se présentent sous forme de molécules : ce sont des éléments nécessaires à la croissance et aux besoins énergétiques des bactéries.

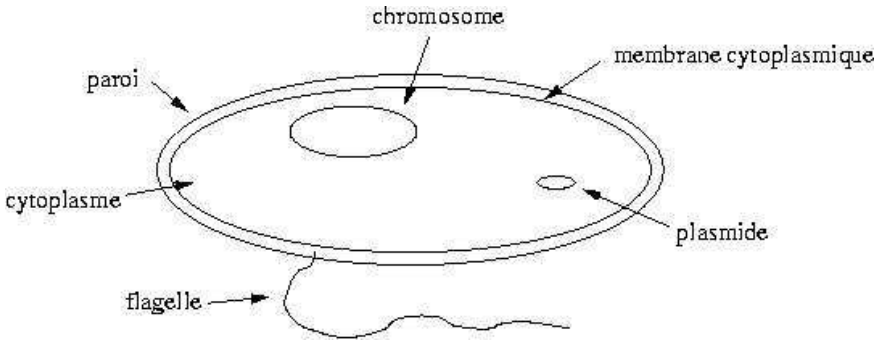


FIG. 1.1 – Schéma simplifié de la structure bactérienne.

molécules, assimilables par la bactérie, sont appelées *substrat*. Au niveau du régime alimentaire, il existe de grandes différences entre bactéries : les bactéries autotrophes sont par exemple capables d'élaborer tous leurs constituants chimiques à partir de composés inorganiques simples, alors que les bactéries parasites requièrent la présence de molécules organiques complexes. Cela se traduit par de grosses différences au niveau de leur répertoire de gènes et de grandes différences dans leur capacité d'adaptation.
D'un point de vue physique, l'architecture d'une bactérie est relativement simple (figure 1.1). La cellule est tout d'abord entourée d'une paroi qui lui donne sa forme, sa résistance, et qui entoure une seconde enveloppe plus ou moins épaisse, la membrane cytoplasmique. C'est une frontière semi-perméable entre la cellule et le milieu environnant : elle est en effet perméable à l'oxygène, au dioxyde de carbone, et à l'eau, mais imperméable aux molécules du milieu tels que sucres, acides aminés, et bien sûr plus encore aux macromolécules. Le (ou les) chromosome(s) et éventuellement le (ou les) plasmide(s) (cf 1.1.5 Structure du génome) se trouvent dans l'environnement délimité par la membrane cytoplasmique : le cytoplasme. Du fait de leur paroi rigide, la plupart des bactéries sont incapables de mouvement. Certaines d'entre elles peuvent toutefois se déplacer grâce à des flagelles dont le nombre varie de un à trente selon les espèces.

1.1.2 L'ADN, support de l'information génétique

L'ADN est une molécule, dont la structure en "double hélice" maintenant bien connue, fut découverte par Watson et Crick (1953). Elle est formée par une succession de petites unités appelées nucléotides, constitués de trois éléments : une molécule d'acide phosphorique, un sucre et une base organique. Dans le cas de l'ADN, le sucre est du désoxyribose ... d'où son nom : Acide(phosphorique) Désoxyribo(se) Nucléique. Les bases, quant à elles, sont au nombre de quatre : l'adénine (A), la thymine (T), la cytosine (C), et la guanine (G). Une molécule d'ADN est habituellement formée de deux chaînes – ou *brins* – de nucléotides. Ces chaînes ont trois propriétés essentielles :

– elles sont *hélicoïdales* : les deux chaînes d'ADN présentent dans l'espace une configuration hélicoïdale. En général, elles s'enroulent autour d'un axe en formant une double hélice droite,

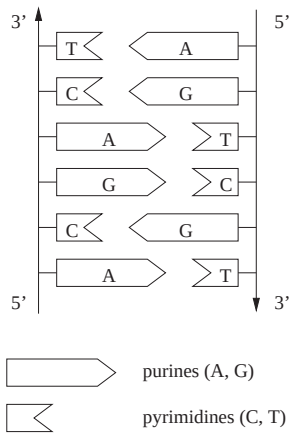


FIG. 1.2 – Les deux chaînes de nucléotides d'une molécule d'ADN. Les paires de bases constituant chaque barreau ont la même taille (une pyrimidine associée à une purine). Les brins sont orientés de 5' vers 3'.

- elles sont *antiparallèles* : les deux brins de nucléotides sont parallèles mais dans des directions opposées (de 5' vers 3'),
- elles sont *complémentaires* : les bases des deux brins ne pouvant pas s'apparier n'importe comment, un peu comme pour les pièces d'un jeu de Lego, les deux bases C et T, dites bases pyrimidiques, forment de petites briques alors que les deux bases A et G, dites bases puriques, forment de grandes briques. On ne peut associer qu'une petite brique - une pyrimidine - avec une grande brique - une purine : l'association de deux purines prendrait trop de place et deux pyrimidines seraient trop éloignées pour former un barreau - une liaison stable. Ceci pourrait laisser supposer qu'en face de A (base purique), on peut trouver aussi bien C que T (base pyrimidique). Il n'en est rien : pour des raisons chimiques (liaisons hydrogène) les couples autorisés sont A-T et C-G (figure 1.2).

Les deux chaînes étant complémentaires quant à l'ordre des bases, chacune des chaînes peut servir de modèle pour fabriquer l'autre. Ainsi, connaissant par exemple la séquence ${}_{5'}\text{ACGACT}_{3'}$ sur l'un des brins, on peut en déduire la chaîne associée sur l'autre brin - le brin complémentaire – qui se lit de droite à gauche : ${}_{3'}\text{TGCTGA}_{5'}$.

L'ordre dans lequel sont répartis les nucléotides dans un segment d'ADN forme un code basé sur l'alphabet de quatre lettres A, T, C, G. En considérant ce code par groupe de trois lettres, on forme de nouveaux mots, les codons. On obtient ainsi 64 mots possibles (4^3) dont 61 codent pour des acides aminés (table 1.1).

L'ADN est porteur d'une information : il contient des suites consécutives de mots que sont les codons. On peut remarquer que, le codon étant constitué de trois lettres, et ne connaissant pas précisément l'emplacement du nucléotide de départ sur le brin, il existe trois cadres de lecture différents. Par exemple, dans AATGCGC, on peut lire **AAT**GCGC - codant pour l'asparagine - en cadre de lecture 1, A**ATG**CGC - codant pour la méthionine - en cadre de lecture 2, et AA**TGC**GC - codant pour la cystéine - en cadre de lecture 3.

Ces séquences d'ADN peuvent avoir plusieurs significations :

Acides aminés	Abréviations		Codons					
Alanine	Ala	A	GCA	GCG	GCT	GCC	–	–
Arginine	Arg	R	CGA	CGG	CGT	CGC	AGA	AGG
Asparagine	Asn	N	–	–	AAT	AAC	–	–
Acide aspartique	Asp	D	–	–	GAT	GAC	–	–
Cystéine	Cys	C	–	–	TGT	TGC	–	–
Glutamine	Gln	Q	CAA	CAG	–	–	–	–
Acide glutaminique	Glu	E	GAA	GAG	–	–	–	–
Glycine	Gly	G	GGA	GGG	GGT	GGC	–	–
Histidine	His	H	–	–	CAT	CAC	–	–
Isoleucine	Ile	I	ATA	–	ATT	ATC	–	–
Leucine	Leu	L	CTA	CTG	CTT	CTC	TTA	TTG
Lysine	Lys	K	AAA	AAG	–	–	–	–
Méthionine (Start)	Met	M	–	ATG	–	–	–	–
Phénylalanine	Phe	F	–	–	TTT	TTC	–	–
Proline	Pro	P	CCA	CCG	CCT	CCC	–	–
Sérine	Ser	S	TCA	TCG	TCT	TCC	AGT	AGC
Thréonine	Thr	T	ACA	ACG	ACT	ACC	–	–
Tryptophane	Trp	W	–	TGG	–	–	–	–
Tyrosine	Tyr	Y	–	–	TAT	TAC	–	–
Valine	V	GTA	GTG	GTT	GTC	–	–	–
Stop	–	–	TAA	TAG	TGA	–	–	–

Tab. 1.1 – Le code génétique

– il peut s'agir de séquences, appelées gènes, et codant pour des protéines ou des ARN structuraux,
– de pseudogènes – gènes inactifs qui ne codent pas pour une protéine fonctionnelle –,
– de bactériophages – virus ayant infecté une bactérie et modifié son patrimoine génétique en incorporant ses gènes à ceux de la bactérie –,
– de séquences répétées qui peuvent être groupées sous forme de *tandems*,
– de transposons – séquences d'ADN mobiles qui peuvent se déplacer d'un site à l'autre de l'ADN ; à chaque extrémité des transposons on retrouve des séquences répétées –,
– ou encore de séquences régulatrices qui jouent un rôle lors de la transcription et de la traduction des gènes en protéines.

L'ensemble de ces objets biologiques est contenu dans le *génome* : ensemble du matériel génétique contenu dans la cellule et se présentant sous la forme d'une ou plusieurs molécules d'ADN. Le génome des êtres vivants est obtenu par séquençage. Les méthodes développées sont de plus en plus efficaces et productives. Ainsi, au début de ma thèse, il y a trois ans, pouvait-on compter tout au plus une vingtaine de génomes bactériens entièrement séquencés. A l'heure actuelle la banque de données GenBank en contient 186. La taille de ces génomes varie fortement, pouvant aller de 300 gènes à plus de 6000. Par exemple, le génome de *Mycoplasma genitalium* est le plus petit génome connu pour une cellule autonome : 580000 bases définissant 484 gènes. La taille moyenne des gènes des génomes séquencés se situe entre 900 et 1000 bases. Chez les procaryotes, le génome est codant à plus de 90% contre moins de 10% chez les vertébrés. Pour être utilisée, cette information génétique doit suivre différentes étapes afin d'obtenir un produit rattaché à une fonction cellulaire.

1.1.3 Biosynthèse des protéines

La synthèse des protéines depuis l'ADN se compose de deux étapes (avec parfois une étape supplémentaire de maturation – modifications post-traductionnelles) : la transcription et la

traduction. Lors de la transcription, l'information portée par l'ADN va en être retranscrite[10] sur un ARN messager ou ARNm. L'ARN - acide ribonucléique - se compose comme l'ADN d'une molécule d'acide phosphorique, d'un sucre et d'une base. Toutefois, le sucre est ici du ribose et la base T (Thymine) est remplacée par l'Uracile, notée U. On parle d'ARN "messager" car il porte une partie de l'information génétique contenue au niveau de l'ADN jusqu'à la machinerie qui synthétisera les protéines. La transcription n'est pas globale : seules de petites portions du génome sont transcrites à un moment donné, variant en fonction de l'environnement par exemple. Ce processus est initié et se termine en deux points précis de l'ADN – les séquences promotrices et le site de terminaison –, l'espace entre les deux constituant une unité de transcription. Un seul brin d'ADN est transcrit : en effet, le gène, en tant qu'information codant pour une protéine, possède un sens de lecture : on parlera de "sens codant" et de "sens inverse" suivant sur quel brin l'information sera lue (le sens de lecture de chacun des brins étant bien sûr opposé). La transcription (figure 1.3) est assurée par une ARN polymérase, une enzyme qui permet de souder des nucléotides les uns aux autres pour former l'ARNm.

La traduction (figure 1.4) se passe au niveau des ribosomes, qui constituent, avec les ARN de transfert – notés ARNt – la machinerie qui convertit les séquences d'ARNm en séquences d'acides aminés dans les protéines. Les ARNt sont responsables du transport des acides aminés jusqu'aux ribosomes et chacun d'eux transporte un acide aminé spécifique. La séquence d'un ARNt comporte un anticodon qui reconnaît le codon correspondant à l'acide aminé qu'il transporte. Un ribosome comporte trois sites de liaison pour les molécules d'ARN – un site pour l'ARNm et deux sites pour les ARNt :

- Un site, appelé site P, fixe la molécule d'ARNt qui est liée à l'extrémité en croissance de la chaîne polypeptidique.
- Un autre site, appelé site A, fixe la molécule d'ARNt entrante chargée d'un acide aminé.

Une molécule d'ARNt n'est solidement fixée à l'un ou à l'autre de ces sites, que si son anticodon s'apparie avec un codon complémentaire sur la molécule d'ARNm qui est liée au ribosome. Les sites A et P sont si proches l'un de l'autre que les deux molécules d'ARNt sont contraintes de s'apparier à des codons adjacents de la molécule d'ARNm. On peut considérer que le mécanisme d'élongation de la chaîne polypeptidique sur un ribosome est un cycle de trois étapes distinctes :

- L'*initiation* : le signal du début de traduction, contenant le codon initiateur, est détecté par un complexe constitué d'un facteur d'initiation (facteur sigma), de la petite sous-unité du ribosome et d'un ARNt caractéristique de l'initiation – ARNti – chargé en formylméthionine. La grosse sous-unité vient alors s'associer à ce complexe pour former un ribosome opérationnel comportant l'ARNti au site P.
- L'*élongation* : un nouvel ARNt chargé de son acide aminé se lie au site A libre du ribosome (adjacent à un site P occupé) en s'appariant aux trois nucléotides de l'ARNm (codon) exposés au site A. La chaîne polypeptidique en cours de synthèse est séparée de la molécule d'ARNt du site P et ajoutée par une liaison peptidique à l'acide aminé fixé à la molécule d'ARNt du site A. Le site P est libéré et l'ARNt lié à la chaîne est transloqué du site A au site P pendant que le ribosome avance exactement de trois nucléotides le long de la chaîne d'ARNm.
- La terminaison se fait par la fixation d'un facteur de "libération" au site A en regard d'un

[10] Biosynthèse d'ARN qui repose sur la complémentarité des bases.

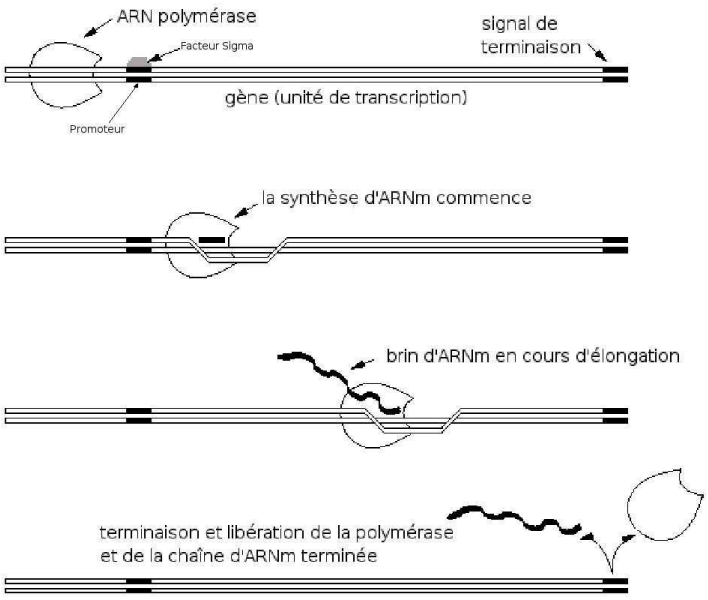


FIG. 1.3 – La transcription de l'information génétique. Les séquences promotrices, TATAAT en position −10 (10 nucléotides avant le site d'initiation de la transcription) et TTGACA et position −35, sont reconnues par le facteur sigma impliqué dans la transcription de la majorité des gènes ; la molécule d'ARN polymérase accomplit sa tâche le long du gène, fabriquant un brin d'ARNm dans lequel chaque base d'ARN est complémentaire de chaque base d'ADN. Il existe deux principaux mécanismes de terminaison, l'un appelé "Rho dépendant" et l'autre "Rho indépendant" : dans le premier cas l'arrêt de la transcription est induit par la présence d'un complexe protéique alors que dans le second cas c'est la formation d'une structure secondaire (motif en tige boucle) au niveau de l'ARNm en cours de synthèse qui conduit à l'arrêt de la transcription.

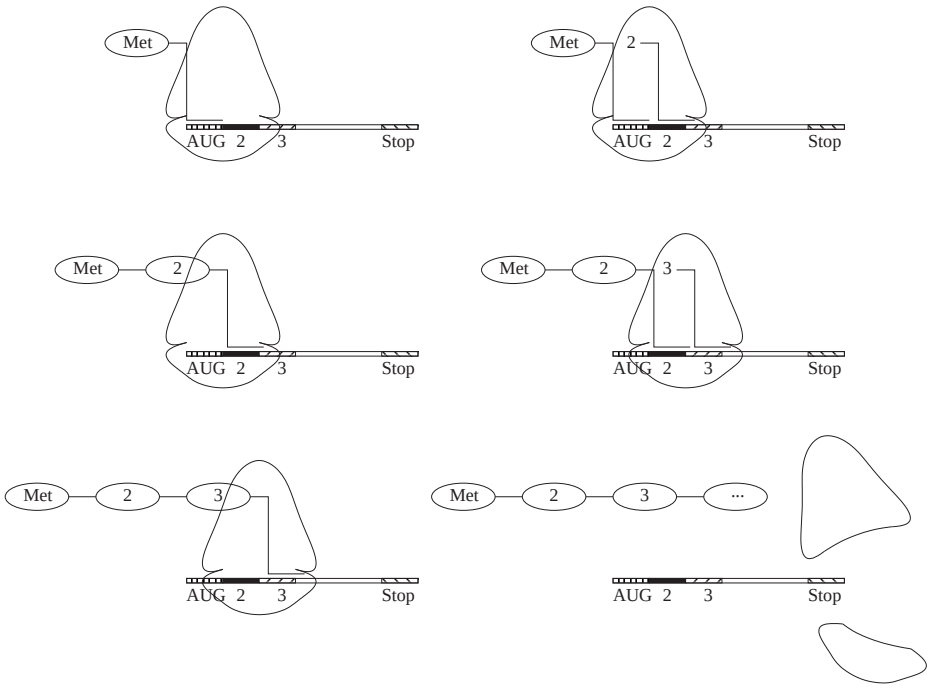


FIG. 1.4 – Les différentes étapes de la traduction : initiation, élongation, et terminaison. Le ribosome se fixe sur l'ARNm. Au fur et à mesure que les codons sont lus sur l'ARNm, un ARN de transfert, porteur de l'acide aminé correspondant au codon, vient se rajouter au complexe Ribosome-ARN. Il crée ensuite une liaison peptidique entre cet acide aminé et la chaîne peptidique en cours d'élongation. Lorsque le ribosome rencontre le codon STOP, il se décroche de l'ARNm et libère la protéine synthétisée.

codon stop. Le ribosome se détache et libère la chaîne polypeptidique.

Ces processus de transcription et de traduction sont sujet à une régulation fine, principalement au niveau de l'initiation. D'autres points de contrôle existent, notamment au niveau de la stabilité des ARNm et des protéines. Ces différents niveaux de régulation permettent à la bactérie d'ajuster la synthèse des protéines en fonction de son cycle cellulaire et de ses besoins physiologiques.

1.1.4 Unités de transcription et de régulation

La régulation de l'expression des protéines se fait essentiellement au niveau de la transcription des ARNm. Je détaillerai ici simplement la découverte de Jacob et Monod (1961) qui conduisit au concept d'opéron. Un opéron est un ensemble consécutif de gènes dont le fonctionnement est contrôlé par une protéine de répression ou d'induction dont la synthèse dépend d'un gène régulateur se trouvant dans la plupart des cas en amont de l'opéron. Généralement les gènes présents dans un même opéron codent pour des protéines impliquées dans un même processus

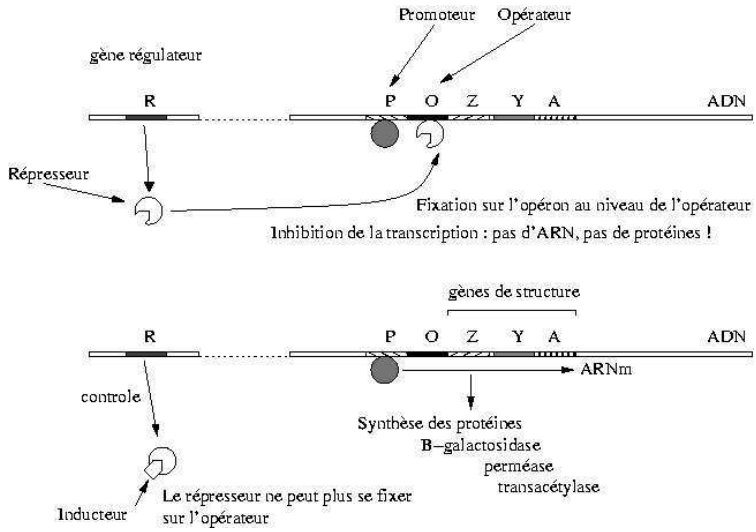


FIG. 1.5 – Modèle de l'opéron lactose d'après Jacob et Monod (1961).

physiologique. Je reprendrai ici l'exemple le plus cité dans la littérature puisqu'à l'origine du concept, l'opéron lactose de *Escherichia coli*. Cette bactérie utilise le glucose comme source d'énergie et pour assimiler ce substrat elle va synthétiser des enzymes capables de le transporter et de le dégrader. En carence de glucose, si du lactose est présent, la bactérie devra assimiler le lactose. C'est donc pour elle essentiel que de synthétiser des enzymes pouvant hydrolyser le lactose en glucose et galactose. Sur la figure 1.5, nous pouvons voir l'action du répresseur sur la synthèse des gènes (l'ensemble O-Z-Y-A constitue une unité génétique à expression coordonnée ou opéron). L'opéron lactose est sous le contrôle du répresseur LacI qui bloque l'expression des enzymes *lac* intervenant dans le métabolisme du lactose (si le lactose et le glucose sont présents dans le milieu, il n'y a pas de dégradation du lactose tant que tout le glucose n'a pas été utilisé). Il s'agit d'un modèle de régulation par induction[11] : la répression est permanente mais peut être levée sur commande par un inducteur.
Par rapport à ce premier niveau de régulation qu'est l'opéron, il y a le *régulon* qui est un ensemble d'opérons contrôlés par le même régulateur et le *stimulon* qui est un ensemble d'opérons répondant au même stimulus de l'environnement quel que soit le mécanisme de régulation mis en jeu. L'ensemble de ces phénomènes peut se représenter sous la forme d'un réseau dont les sommets sont des objets biologiques et les arêtes les interactions existant entre ces objets.

1.1.5 Structure du génome

Les bactéries n'ont qu'un (plus rarement plusieurs) chromosome, qui est généralement circulaire et bien plus petit que les chromosomes linéaires des eucaryotes. Une des particularités des procaryotes est la présence éventuelle d'un ou plusieurs plasmides, petits morceaux d'ADN

[11] L'opéron tryptophane est lui un modèle de régulation par répression

circulaire qui se trouvent à côté du chromosome, qui sont indépendants de l'ADN principal, et qui ne sont généralement pas essentiels au métabolisme de la cellule hôte. Ce sont ces chromosomes et plasmides qui portent l'information génétique. Pour être transmise aux cellules filles lors de la division cellulaire, cette information va être répliquée. Ce processus est semi-conservatif : dans les deux brins d'une molécule d'ADN, il y a toujours un brin ancien et un brin nouvellement formé. Le point de démarrage de la réplication se situe au niveau de l'*origine de réplication* : un processus complexe sépare les deux brins d'ADN au niveau de cette origine, met en place les premières amorces, et démarre des fourches de réplication dans les deux sens[12]. La synthèse ne pouvant s'effectuer que dans le sens 5' vers 3', la réplication se déroulera en continu sur un brin – dit brin "avancé" – et en discontinue sur l'autre brin – dit brin "retardé".

1.2 Evolution des génomes

La reproduction des bactéries prend la forme d'un procédé asexué où chacune grandit en taille, duplique son chromosome, puis se divise en deux bactéries identiques, ou clones. Malgré ce type de reproduction clonal, qui laisserait supposer qu'une bactérie fille soit identique à sa "mère", l'information génétique est modifiée d'une génération à l'autre. On peut par exemple s'apercevoir qu'une bactérie pathogène devient résistante aux antibiotiques qui la combattent.

1.2.1 Remaniements chromosomiques

Le premier des mécanismes de variabilité est le remaniement chromosomique. Il s'agit de séquences d'ADN qui, généralement, lors du processus de réplication de la bactérie, se dupliquent et s'insérent dans l'ADN. La séquence peut s'insérer sur le même brin ou sur le brin complémentaire, ce qui aura pour effet de modifier l'orientation des gènes qu'elle porte. Plusieurs remaniements apparaissant sur une même séquence peuvent modifier l'ordre des gènes. En partant par exemple de la séquence $S = a\ b\ c\ d\ e$, où a, b, c, d, et e sont des gènes dont l'orientation est indiquée par le signe les précédant (une absence de signe indique un sens codant et un "-" indique un sens inverse), il y a duplication de la séquence $b\ c\ d$: $S = a\ b\ c\ d\ \underline{b\ c\ d}\ e$. Suite à cette duplication, il se produit une inversion de $c\ d\ e$: $S = a\ b\ c\ d\ b\ \underline{-e\ -d\ -c}$. On aboutit ainsi à une modification de l'ordre des gènes dupliqués. En multipliant ces opérations, les génomes bactériens sont extrêmement remaniés. De plus, d'autres mécanismes de variabilité entrent en jeu.

1.2.2 Mutations

Les informations contenues dans l'ADN et nécessaires à la synthèse des différentes protéines vont être transmises de génération en génération par le processus de réplication semi-conservative de cet ADN. Dans chaque cellule fille issue d'une réplication, la double hélice d'ADN est composée d'un brin nouvellement synthétisé et du brin matrice. Lors de ce processus où l'ADN résultant devrait être l'exacte réplique de son parent, il peut y avoir des ratés à cause d'accidents de copie

[12] La réplication est bidirectionnelle, c'est-à-dire qu'elle se prolonge simultanément à gauche et à droite du point d'initiation.

des bases puriques ou pyrimidiques. Par ce moyen l'ADN peut muter. Ces mutations peuvent être de trois types :

- la substitution : une base est mal copiée.
- la délétion : une base est oubliée.
- l'insertion : une base est ajoutée.

Ces accidents de copie sont tout à fait analogue à une faute de frappe qui se produirait lors de la copie d'un texte. Prenons par exemple deux phrases : d'un côté "La **r**eine s'en est allée." et de l'autre "La **p**eine s'en est allée.". Ces deux phrases bien que sémantiquement et syntaxiquement correctes n'ont pas le même sens. Une seule lettre a été modifiée entre les deux et dans un cas le chagrin s'est évanoui, alors que dans l'autre la femme du roi est partie. Au niveau de l'ADN, les mutations peuvent avoir ou pas une influence sur le gène (et donc sur la protéine). Ces mutations se différencient en deux groupes :

- Les mutations sans changement du cadre de lecture (substitutions) :
  - Les mutations "silencieuses" : La substitution de nucléotide est sans effet : le codon qui en résulte code le même acide aminé. Par exemple, si UU**U** est remplacé par UU**C**, ces deux codons codent pour la Phenylalanine ; cette substitution n'a aucune conséquence quant à la séquence protéique.
  - Les mutations "conservatrices" : Le codon d'un acide aminé est remplacé par le codon d'un acide aminé du même groupe. Par exemple, si A**A**A (Lysine) est muté en A**G**A (Arginine), ces deux acides aminés appartiennent au même groupe (acides aminés basiques) et la mutation n'aura le plus souvent aucune conséquence.
  - Les mutations portant sur le codon Stop : Cette mutation transforme un codon quelconque en codon Stop, raccourcissant la taille du gène. De même, un codon Stop peut être muté en un codon quelconque, allongeant la taille du gène.
- Les mutations avec changement du cadre de lecture (insertions et délétions) : Ces mutations sont dues à l'insertion ou à la délétion d'une ou plusieurs bases qui entraînent un décalage dans la lecture des triplets. Si cette mutation, et donc le déphasage, se produit au début du gène, le gène en question sera totalement modifié. Prenons un exemple où nous effectuerons deux délétions successives :

Séquence initiale	ATG	**G**CC	TCT	AAC	TAA
	Met	Ala	Ser	Asn	Stop
Décalage de 1	ATG	**C**CT	CTA	ACT	AA
	Met	Pro	Leu	Thr	–
Décalage de 2	ATG	CTC	TAA	CTA	A
	Met	Leu	Stop	Thr	–

Le gène codant initialement pour quatre acides aminés ne code plus que pour deux.

1.2.3 Transfert horizontal

L'héritage vertical de matériel génétique d'une génération à une autre (les deux cas précédents) est le mode de transmission le plus courant à l'intérieur des organismes vivants. Dans certaines circonstances, il arrive que du matériel génétique se transfère entre espèces éloignées. On parle alors de *transfert horizontal*. Trois mécanismes principaux peuvent intervenir :

- La *transformation* est le mécanisme le plus simple. Dans le milieu extérieur se trouve de

l'ADN libre qui résulte en général de la mort d'un organisme. Cet ADN libre peut être intégré à l'intérieur d'une cellule particulière, puis intégré au génome de la bactérie. Il y a donc transfert horizontal entre deux espèces qui peuvent être tout à fait différentes.

- La *conjugaison* : les organismes ont mis au point un système leur permettant de s'échanger du matériel génétique, en particulier des plasmides (Christie *et col.*, 1987). Le processus est le suivant : deux cellules entrent en contact et s'échangent toute une partie de leur matériel génétique. Ce mécanisme est particulièrement important à l'intérieur d'une même espèce, mais le phénomène de conjugaison peut également avoir lieu entre des organismes qui ne font pas partie de la même espèce.
- La *transduction* : l'ADN est transféré d'une espèce à une autre via des virus bactériophages. Certains virus sont des organismes capables d'intégrer leur matériel génétique dans l'hôte et utilisent la machinerie transcriptionnelle et traductionnelle de l'hôte pour donner naissance à de nouvelles particules virales. Ils peuvent amener par erreur une partie du matériel génétique de l'hôte et comme ces organismes n'ont pas une spécificité d'hôte très importante, ils sont capables de passer d'une espèce à une autre et donc de transférer du matériel génétique par erreur d'une espèce à une autre. Le virus ne peut pas dans ce cas infecter l'hôte car il n'a plus tout son matériel génétique. Ce transfert horizontal est bénéfique pour l'hôte car il n'est pas victime du virus mais il est très limité, du point de vue écologique et évolutif, par la spécificité de l'hôte et par l'efficacité de la recombinaison ((Birge, 1994), (Matic, 1995)).

Ces mécanismes sont très fréquents chez les procaryotes et génèrent l'échange de beaucoup de matériel génétique. Néanmoins, ce matériel génétique n'est pas forcément conservé par l'organisme suite à un transfert horizontal : il faut en effet que le gène, ou le complexe de gènes, transféré horizontalement soit avantageux pour l'organisme. Dans la plupart des cas, il n'y aura pas d'avantage sélectif et la modification se perdra au cours de l'évolution. Dans d'autres cas plutôt rares, il y aura acquisition d'une nouvelle fonction ou d'une résistance aux antibiotiques. Dans ce cas là, le gène transféré est conservé dans la population car il y a un avantage sélectif. Je rappelerai ici qu'une mutation n'est pas toujours néfaste (elle est même bien souvent neutre), c'est d'ailleurs une des bases de la "théorie de l'évolution". Si la mutation représente un avantage – une meilleure adaptation à l'environnement, les individus mutés et leurs descendants survivront mieux que les individus non mutés qu'ils finiront par remplacer.

La détermination de la séquence complète – et donc du séquençage – d'un génome n'est que la première étape de son étude. Il est en effet nécessaire d'effectuer une *annotation*, c'est-à-dire de déterminer exactement où se situent les gènes et leurs régions régulatrices.
La connaissance des génomes de plusieurs organismes permet d'une part de faciliter l'identification des gènes via des comparaisons entre séquences génomiques et, d'autre part, de comparer les gènes eux-mêmes. Ces recherches qui peuvent être menées sur des gènes présents dans des organismes phylogénétiquement très distants, permettent de mieux cerner la fonction et l'importance de ces gènes ainsi que leur histoire évolutive.

1.3 Génomique comparative

Pour comparer les génomes, il faut pouvoir créer des liens entre gènes appartenant à des génomes différents. Pour cela on a recours à la théorie de l'évolution ; comprendre l'évolution consiste à déterminer les filiations et à déceler les mécanismes qui ont présidé à la structuration actuelle de la biodiversité, en permettant les multiples adaptations des animaux, végétaux, et organismes unicellulaires. Il s'agit donc de décrypter la manière dont les organismes se sont modelés tant d'un point de vue structurel que fonctionnel, et de rechercher l'origine des nouveautés et des potentialités évolutives. Ainsi, les recherches sous-tendues par le concept d'évolution se séparent-elles en deux parties : les unes visent la reconstruction de l'histoire de la vie, et les autres cherchent à comprendre les modalités et les processus de l'évolution ((Guyader, 2003), (Janvier, 2003)).

1.3.1 Relations d'homologie

Le concept fondamental de l'évolution est l'*homologie*. C'est Etienne Geoffroy Saint-Hilaire qui, le premier (1843), définit l'homologie, notion essentielle permettant de comparer des organismes ayant le même plan d'organisation. Deux organes sont alors dits homologues s'ils ont la même situation dans un plan d'organisation : deux organes homologues peuvent n'avoir ni la même taille, ni la même forme, ni la même fonction. Pour Darwin, ce sont ces caractères homologues, hérités d'un ancêtre commun, qui sont utiles en taxonomie.
L'histoire de l'évolution des êtres vivants, et donc les relations de parenté entre les espèces et les *taxons* (groupes d'espèces nommés, comme par exemple les eubactéries et les archebactéries), peut être illustrée en suivant un modèle d'arbre. Cette représentation arborée est à la fois simple et en deux dimensions. On distingue deux types de représentations (d'après Darlu et Tassy (1993)) :

- le *cladogramme*, un diagramme ramifié où tous les taxons sont placés à l'extrémité des branches. Il s'agit d'un arbre de parenté construit à partir du principe de parcimonie : les taxons sont reliés sur la base de leur ressemblance maximale en terme d'homologie.
- l'*arbre phylogénétique*, qui apporte les mêmes informations que le cladogramme mais auquel on peut en outre adjoindre une échelle de temps.

Cette approche phylogénétique permet une vision synthétique de l'ascendance commune des taxons et peut être interprétée de la manière suivante : l'arbre qui maximise les homologies est l'arbre de longueur minimale, celui qui contient le minimum de transformations, autrement dit le minimum de pas évolutifs. On obtient l'arbre le plus parcimonieux : il s'agit du *principe de parcimonie*[13]. Les états définis par un noeud sont interprétés comme présents chez le dernier ancêtre commun des taxons qui dérivent de ce noeud. La branche qui relie deux noeuds internes est le lieu des transformations évolutives. La longueur d'une branche correspond au nombre de transformations ainsi optimisées (Barriel et Tassy, 2003).

[13]Il faut noter l'existence d'autres méthodes de choix d'un arbre comme par exemple l'approche probabiliste suivant le *principe de vraisemblance* : on choisit l'arbre le plus vraisemblable, celui dont le produit des probabilités est maximal.

On peut distinguer trois sous-types d'homologie (Fitch, 2000) :

- L'*orthologie* où les séquences sont issues d'un événement de spéciation ; on constate généralement une conservation de la fonction des gènes.
- La *paralogie* où les séquences sont issues d'un événement de duplication ; on constate générallement une spécialisation de la fonction des gènes.
- La *xenologie*, ou *transfert horizontal*, où il y a un transfert de matériel génétique étranger.

Un récapitulatif de ces relations est proposé en figure 1.6. Rappelons que d'après Fitch (2000), un gène ne possède pas forcément un unique orthologue dans un génome donné. Sur la figure 1.6, le gène A est orthologue à B_1, B_2, C_1, C_2 et C_3. Mais, lorsque des gènes orthologues possèdent un ou des paralogues (comme par exemple B_1 et C_1 qui sont orthologues et C_1, C_2 et C_3 qui sont paralogues), présentant des trajectoires évolutives différentes[14], il est possible que l'un d'eux soit plus proche du point de vue de la similarité du gène orthologue et non paralogue (B_1). On parlera alors d'*isorthologues* (B_1 et C_1 sont par exemples isorthologues car le taux de similarité entre C_1 et C_2 ou entre C_1 et C_3 est inférieur à celui entre B_1 et C_1). L'isorthologie présente un intérêt lors de la comparaison des génomes car elle permet de faire des inférences fonctionnelles plus précises : en absence de duplication récente des deux gènes orthologues B_1 et C_1, il y a de grandes chances pour qu'ils aient conservé la même fonction. Plus formellement :

Définition 3.1 *Soient quatre gènes $A1$, $A2$ du génome A, et $B1$, $B2$ du génome B. Considérons que $A1$ et $A2$ sont paralogues dans A et $B1$ et $B2$ sont paralogues dans B. Alors $A1$ et $B1$ sont dits isorthologues si et seulement si aucune des paires de paralogues n'est à une distance évolutive plus faible que les gènes $A1$ et $B1$. Cette définition a été volontairement simplifiée : elle reste valable pour n gènes paralogues dans A et m gènes paralogues dans B.*

1.3.1.1 Comment retrouver l'information évolutive

La reconstruction de la trajectoire évolutive d'un gène peut poser de nombreux problèmes : les génomes bactériens sont extrêmement remaniés, ce qui peut conduire à des délétions de gènes ; les séquences ont tellement évolué qu'il n'y a plus de signal phylogénétique (problème de saturation), ou encore, il y a eu des transferts horizontaux. Il ne faut pas non plus oublier que les méthodes de reconstruction de la trajectoire évolutive d'un gène (distances, reconstruction d'arbre, ...) sont parfois imprécises. En guise d'exemple, nous pouvons considérer le scénario suivant : tout d'abord il y a duplication d'un gène, puis, suite à un evènement de spéciation, la première espèce perd l'une des copies et la seconde l'autre copie (figure 1.7). Il y a donc perte de l'information sur la trajectoire évolutive de ce gène : A et C vont donc être considérés comme isorthologues, alors qu'ils sont paralogues.
Les transferts horizontaux sont également très difficile à détecter : quand un gène passe d'une espèce à une autre, la situation est idéale pour avoir des altérations de la vitesse évolutive, et ce en raison de l'adaptation du gène à son nouvel environnement. En général, un gène transféré évolue plus vite qu'un gène non transféré, ce qui pose de gros problèmes de détection de l'orthologie.

[14] Evolution par mutation par exemple.

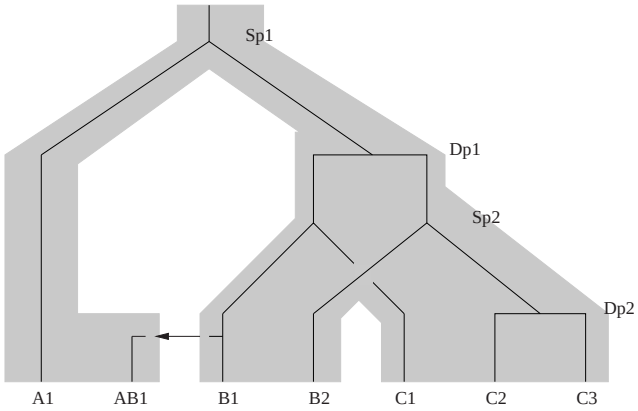


FIG. 1.6 – Evolution d'un gène depuis un ancêtre commun jusqu'à trois espèces différentes A, B et C. Les événements de duplication sont notés Dp1 et Dp2, et les événements de spéciation sont notés Sp1 et Sp2. Tous ces gènes sont homologues. $AB1$ est xenologue, $C2$ et $C3$ sont paralogues, et $B1$ et $C1$ sont orthologues, ... (d'après Fitch (2000))

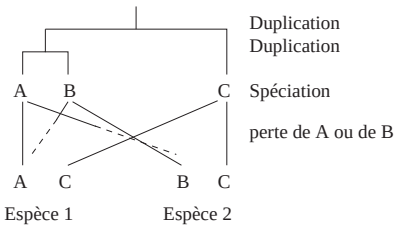


FIG. 1.7 – La perte d'un gène : l'espèce 1 perd le gène B, et l'espèce 2 perd le gène A : perte d'information sur la trajectoire évolutive de ce gène.

L'isorthologie, quant à elle, qui devrait être en théorie transitive, ne l'est malheureusement pas du fait des différents problèmes énumérés ci-dessus. En effet, nous pouvons avoir un gène A_1 du génome A isorthologue à B_1 du génome B, lui-même isorthologue à C_1 de C. Mais C_1 sera peut-être plus proche d'un gène A_2 de A qu'il ne l'est de A_1. Dans le cas des gènes de transporteurs ABC, j'ai calculé le nombre de fois où la transitivité n'était pas respectée dans la base ABCdb (Quentin et Fichant, 2000) : pour chaque gène, il s'agit du nombre de triangles que l'on peut former avec d'autres gènes en suivant une relation d'isorthologie divisé par le nombre de triangles possibles. On obtient ainsi un taux de 29% de gènes pour lesquels la transitivité n'est pas respectée. Ce phénomène est donc loin d'être rare en considérant cette seule famille génique.

1.3.1.2 Identification des relations d'homologie

Dans sa définition moderne, l'homologie est la relation entre deux caractères – gènes ou organes – descendant, généralement après divergence, d'un ancêtre commun. Cette définition, assez peu contraignante, pose quand même un problème : comment reconnaître avec certitude deux séquences homologues ? Le travail s'effectue sur des données contemporaines, ayant évolué au cours du temps, cette relation est donc difficile à détecter. En définissant l'homologie par rapport à la séquence nucléotidique des caractères étudiés, on pourrait par exemple dire qu'elles doivent être identiques à $X\%$ ... mais quelle est la valeur de X ? Lorsque l'on parle d'homologie, on ne peut la quantifier. L'exemple classique, utilisé dans la littérature, est celui de la femme enceinte : une femme est enceinte ou ne l'est pas, elle ne peut l'être à $X\%$. Il en va de même de l'homologie : deux séquences peuvent posséder $X\%$ de résidus identiques mais elles ne seront pas "homologues à $X\%$". Pour identifier les gènes homologues entre espèces on doit avoir recours aux alignements de séquences : les nucléotides correspondant à des positions homologues dans les séquences sont mis en vis à vis. Il y a donc possibilité de détecter un signal taxonomique. Un problème se pose lorsque les séquences ont accumulé tant de mutations que l'alignement devient peu robuste (beaucoup d'insertions/délétions) voire même pas significativement différent de ce que l'on obtiendrait avec des séquences sans liens de parentés (saturation). L'idéal serait d'effectuer des alignements globaux (recommandés en phylogénie) mais ces méthodes sont très couteuses en temps de calcul. On utilise donc des méthodes heuristiques plus rapides, basées sur des alignements locaux et où le score n'est pas une distance phylogénétique. J'ai utilisé le logiciel BLAST (pour **B**asic **L**ocal **A**lignement **S**earch **T**ool[15]) qui est basé sur une méthode statistique (Altschul *et col.*, 1990). Il est destiné à trouver les alignements optimaux locaux de meilleur score entre la séquence requête et une banque de séquences. Le score est une valeur permettant de qualifier et de quantifier la similitude entre séquences. Ce score croît tant que l'on trouve des bases identiques successives, et décroît quand elles sont différentes ou que l'on insère ou supprime une base ; quand on arrive à une valeur négative, le score est mis à zéro. Les valeurs des pénalités de score en cas de substitution ou d'absence de similitude au sein d'un espacement sont des paramètres pouvant être fixés par l'utilisateur. L'idée principale de l'algorithme est que les bons alignements doivent contenir quelque part des petits segments strictement identiques ou de score très important. Ces éléments sont des graines où l'alignement est ancré et à partir duquel

[15]Outil de recherche d'alignement local basique.

il s'étend (Altschul *et col.*, 1990). La première version de l'algorithme BLAST ne permet ni insertion ni délétion, mais est très rapide et attribue une valeur statistique au score obtenu. Cet algorithme a été modifié pour donner le jour à plusieurs autres versions, dépendant de différents besoins : BLAST2 permet ainsi les insertions et les délétions, et PSI-BLAST est une version qui construit des motifs à partir d'alignements itératifs (Altschul *et col.*, 1997). Finalement, la signification des segments similaires est évaluée statistiquement. Celle-ci est faite en fonction de la longueur et de la composition de la séquence et de la taille de la banque. Cette estimation donne en fait la probabilité que l'on a d'observer au hasard une similitude de ce score à travers la banque de séquences considérée. On peut aussi noter que des filtres ont été conçus pour masquer les régions de faible complexité qui conduisent à des résultats statistiquement significatifs mais sans aucun intérêt biologique, autrement dit, un score élevé n'a pas toujours de signification.

Un point très intéressant de cette méthode, pour la comparaison de protéines, est qu'elle ne recherche pas seulement des zones d'identité mais accepte la présence de similitudes de par l'utilisation d'une matrice de substitution (telle que la Blosum62 pour les protéines (Henikoff et Henikoff, 1992)). Ceci permet d'intégrer directement dans les calculs des critères biologiques. Nous gardons bien sûr à l'esprit que cette méthode basée sur des alignements locaux n'est pas satisfaisante mais c'est une approche qui nous permet d'obtenir des données sur lesquelles travailler, une autre méthode consistant à récupérer des données pour lesquelles les calculs ont déjà été effectués.

1.3.1.3 Acquisition des relations entre gènes issus de différents génomes

Pour comparer des gènes entre génomes, on peut soit faire les calculs d'homologie, soit utiliser une banque de données telle que COG (Tatusov *et col.*, 2001), développée au NCBI, et qui contient des informations sur les liens d'homologie existant entre gènes. Comme une relation d'orthologie indique en général une conservation de la fonction, cette banque permet de prédire la fonction des produits de gènes nouvellement séquencés : les produits de gènes présents au sein d'un même groupe COG sont supposés avoir la même fonction. Pour pouvoir créer ces groupes COG (**C**lusters of **O**rthologous **G**enes – Groupes de gènes orthologues), il faut disposer au préalable des génomes complets et d'une liste complète de leurs orthologues. Chaque groupe COG devrait représenter le résultat de l'évolution d'un gène ancestral unique par une série d'événements de spéciation et de duplication. En d'autres termes, un groupe COG renfermera des orthologues au sens de Fitch (2000). Pour déterminer ces groupes par comparaison de séquences, pour chacun des gènes initiaux on recherche dans un premier temps, à l'aide du logiciel BLASTP (Altschul *et col.*, 1997), le meilleur score avec un gène de chacun des autres génomes mis en jeu.

Notation 3.2 *Pour chaque gène, les meilleurs scores détectés lors de la comparaison avec chacun des génomes cibles (un meilleur score par génome) définissent une relation binaire non symétrique notée BeT pour "**Be**st hi**T**".*

L'identification des COG est basée sur les différents motifs détectés dans le graphe formé par les BeTs (Tatusov *et col.*, 1997). Le motif le plus simple, et le plus important, est le triangle

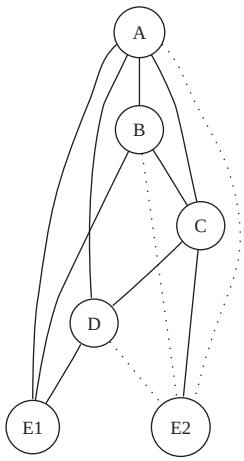


FIG. 1.8 – Construction d'un groupe COG : les arêtes pleines indiquent un BeT symétrique et les arêtes en pointillés indiquent un BeT orienté. Les gènes E1 et E2 appartiennent à la même espèce, les autres gènes A à D étant dans des espèces différentes. On peut remarquer que ce groupe est formé de triangles possédant une arête commune. E1 et E2 sont des paralogues, on notera que E2, bien que ne formant pas de triangle en arêtes pleines (mais en arêtes mixtes), est intégré au groupe COG.

qui est l'empreinte typique d'orthologues. Si un gène A possède des BeTs avec deux gènes B et C issus d'autres génomes, et que ces BeTs sont réciproques (ou symétriques), alors il est hautement probable que ces trois gènes soient orthologues. En effet, si B est le BeT de A et que C est le BeT de B, la probabilité pour que C soit le BeT de A par chance est proche de $1/N_C$ où N_C est le nombre de gènes dans le génome de C (Tatusov *et col.*, 1997). Ce triangle de BeTs représente le groupe COG minimal. Par la suite, la procédure utilisée pour créer les COG est de trouver tous les triangles formés par des BeTs et d'aggréger les triangles qui ont un côté commun, et ce jusqu'à ce qu'aucun nouveau triangle ne puisse être intégré. Les groupes ainsi produits contiennent des orthologues de différentes espèces et aussi, dans certains cas, des paralogues d'une même espèce. La figure 1.8 montre une illustration de ce cas.
A cause de l'existence des paralogues, les BeTs qui forment les triangles ne sont pas nécessairement symétriques. Il faut noter que, dans certains cas, des groupes COG peuvent être morcelés. Si, par exemple, nous sommes en présence d'une protéine multi-domaines[16], il se peut qu'une région appartienne à un COG et qu'une seconde région appartienne à un autre groupe COG. Pour remédier à ce problème, dans chaque protéine multi-domaines on isole alors les domaines individuellement, et une seconde comparaison de séquences est réalisée sur ces domaines. Certains groupes COG peuvent contenir des gènes majoritairement paralogues de plusieurs espèces plutôt que des gènes orthologues, ce phénomène pouvant être induit par la perte d'un gène au cours de

[16]Certaines protéines peuvent posséder plusieurs zones fonctionnelles distinctes (par suite de la fusion de plusieurs gènes par exemple). Ces zones sont appelées des domaines, d'où la dénomination de protéine multi-domaines.

l'évolution. Quand un gène d'une paire de paralogues est perdu dans une espèce mais pas dans les autres, alors deux groupes COG qui étaient distincts peuvent être artificiellement aggrégés. De même, le niveau de similarité entre membres de chaque groupe est analysé et les groupes qui semblent contenir un ou plusieurs groupes COG[17] sont découpés manuellement.
Les groupes COG sont regroupés sur la base de grandes fonctions biologiques. Ces fonctions, au nombre de dix-huit, sont très générales[18]. Les auteurs attirent notre attention sur le fait qu'il existe des groupes COG de taille importante dont les membres possèdent des relations complexes entre eux. Parmi ceux-ci on trouve le groupe des domaines NBD. Les gènes des transporteurs ABC sont répartis dans 173 groupes COG dont la taille varie d'une dizaine à environ 500 gènes.

Ces outils sont utilisés pour l'analyse globale des génomes, pour effectuer des comparaisons de gènes et, également, pour l'annotation des génomes. Il s'agit toujours de prédictions qu'il faudra vérifier par la suite, mais le nombre de génomes séquencés croissant très rapidement, ces outils permettent une première approche générale de ces nouveaux génomes.

1.3.2 Relations de proximité

La disponibilité de multiples génomes entièrement séquencés offre l'opportunité de développer de nouvelles méthodes de prédiction de la fonction des protéines, complémentaires des méthodes traditionnelles basées sur les similarités. D'après Huynen et Snel (2000), ces méthodes se décomposent en trois familles : l'identification des gènes ayant fusionnés (Suhre et Claverie, 2004), les profils phylogénétiques (Enault *et col.*, 2004), et l'analyse de la conservation du voisinage local des gènes. Ces dernières années, les études portent principalement sur la dernière approche. Celle-ci est basée sur l'observation suivante : dans les génomes bactériens, les gènes qui apparaissent de manière répétée dans une même localisation chromosomique – dans des opérons potentiels – codent pour des protéines liées fonctionnellement (e.g. ces protéines font partie du même complexe protéique ou de la même voie métabolique (Mushegian et Koonin, 1996) et (Tamames *et col.*, 1997)). Cette observation est d'ailleurs confirmée par une analyse statistique des régions codant pour des gènes fonctionnellement liés. Ces dernières ont une très forte tendance à être dans le même ordre quand elles sont localisées dans le même voisinage (Overbeek *et col.*, 1999). Dans ce contexte, la question de la conservation d'un contexte génétique est posée à travers l'analyse des gènes appartenant à la même unité transcriptionnelle. Comme ces unités sont généralement inconnues, elles sont inférées en tant qu'ensemble de gènes sur le même brin possédant des régions d'espacement de moins d'une centaine de paires de bases (appelées gènes en *série* – ou *run* – dans (Overbeek *et col.*, 1999)).
La puissance d'une telle analyse est liée à la disponibilité de nombreux génomes taxonomiquement distants entièrement séquencés. En effet, chez les procaryotes, les réarrangements de l'ordre des gènes sur le chromosome sont fréquents et conduisent à une conservation très limitée de l'ordre des gènes sur des distances évolutives relativement grandes ((Dandekar *et col.*, 1998), (Huynen et Bork, 1998)). Snel *et col.* (2000) ont montré que la probabilité pour que deux gènes apparaissent de manière répétée dans une même localisation chromosomique par pur hasard était

[17] Il y a actuellement 4873 groupes COG.
[18] En guise d'exemple, on peut citer la fonction C : conversion et production d'énergie.

très faible. Dans des génomes remaniés aléatoirement, la probabilité pour deux espèces est de 0.02. Pour trois espèces la probabilité est inférieure à 0.002 et pour quatre espèces ou plus, elle est inférieure à 0.0005. Seuls les gènes en *série*, conservés entre des génomes taxonomiquement distants, sont la signature de contraintes sélectives qui pourraient révéler des liens transcriptionnels ou fonctionnels.
Etrangement, plusieurs études ont montré que seulement un petit nombre de gènes était fortement conservé dans un même voisinage lorsque des génomes distants étaient comparés, et que la plupart de ces gènes codent pour des protéines en interaction physique directe. En dehors des modèles basés sur les interactions physiques et la co-régulation, d'autres hypothèses ont été proposées pour expliquer la conservation du voisinage (Lawrence, 1999). Certaines d'entre elles reposent uniquement sur la proximité génomique sans contraintes sur l'orientation ni l'espacement intergénique. Par exemple, un groupement de gènes peut être bénéfique lorsque une forte concentration locale de produits protéiques dans le cytoplasme est nécessaire pour créer une fonction complexe (ce modèle est appelé "Molarity Model" dans Lawrence (1999)). D'un point de vue évolutif, les gènes impliqués dans une même fonction peuvent conférer un phénotype sélectionnable et ainsi, leur regroupement en régions sur le chromosome peut faciliter leur propagation à d'autres organismes au travers de transferts horizontaux. Dans ce cas, le regroupement des gènes est initialement bénéfique aux gènes eux-mêmes, et non à leurs organismes hôtes (ce modèle est appelé "Selfish Operon Model" dans Lawrence (1999)). Il y aurait donc des conservations de gènes dans le même voisinage qui ne suivent pas la contrainte d'appartenir au même opéron (Lathe *et col.*, 2000). En d'autres termes, il y aurait d'autres contraintes que l'organisation en unités de transcription pour rendre compte de la conservation de gènes dans le même voisinage.

L'étude de la conservation de gènes dans une même proximité a conduit à définir la synténie : on dit de deux gènes d'un même organisme qu'ils sont en *synténie* s'ils sont portés par le même chromosome. Etant donné deux organismes A et B et deux gènes en synténie dans A, on dit que cette synténie est *conservée* si les orthologues des deux gènes de l'espèce B sont également en synténie (pour une introduction plus générale (Médigue *et col.*, 2002)). Chez les bactéries qui ne possèdent généralement qu'un seul chromosome, les définitions précédentes ne sont pas vraiment applicables puisque par définition tous les gènes sont en synténie dans une espèce et en synténie conservée entre deux espèces. On parle alors plutôt de *synténie bactérienne* pour indiquer qu'un groupe de gènes possède la même organisation locale dans une espèce A que leurs orthologues dans une espèce B.
Dans de nombreux génomes procaryotes il existe une corrélation très forte entre cette organisation synténique et l'interaction physique entre les produits des gènes voisins (Overbeek *et col.*, 1999). Les groupes de synténie sont définis selon des critères plus ou moins stricts, liés à la similitude des séquences, la conservation de l'ordre des gènes sur le chromosome, la distance respective des gènes les uns par rapport aux autres, etc (von Mering *et col.*, 2003). Une définition formelle des syntons :

Définition 3.3 *Soient n génomes $G_1, ..., G_n$. On considère que le gène g_i possède un orthologue dans chaque génome $G_j, 1 \leq j \leq n$. On appelle synton un ensemble de m gènes ordonnés qui ne sont pas forcément adjacents $\Upsilon = \{g_1, ..., g_m\}$. Les gènes de Υ sont tels qu'il n'existe pas plus de*

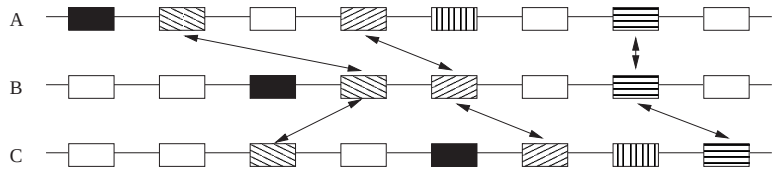


FIG. 1.9 – Groupes de synténie (syntons) conservés entre trois génomes avec un seuil de gènes intercalés $\delta = 2$. Les gènes de même motif sont orthologues.

δ gènes n'appartenant pas à Υ entre les gènes g_i et g_{i+1} pour $1 \leq i \leq m-1$ et ce dans chaque génome.

L'analyse de synténie met en évidence les syntons, ensemble de gènes dont l'organisation chromosomique est conservée entre deux espèces ou plus. Ils vont permettre d'inférer certaines propriétés aux protéines codées. La figure 1.9 montre un exemple de synténie : le synton n'est constitué que de trois gènes car l'ordre des gènes doit être respecté sur les trois génomes et les gènes doivent être orthologues sur les trois génomes. On peut toutefois suspecter le gène sans relation d'orthologie du génome B se situant juste avant le dernier gène du synton d'avoir une fonction proche du gène hachuré verticalement et qui est présent dans les génomes A et C.
Ce type d'analyse permet également d'identifier des mécanismes évolutifs résultant de la fusion/fission de gènes ainsi que des opérons conservés entre différentes espèces.

1.4 Les systèmes intégrés

Les bactéries peuvent se développer dans des environnements très variés grâce à leur capacité d'adaptation à l'environnement. Les réponses apportées aux variations de l'environnement peuvent aller de la modification de l'expression de quelques gènes à la mise en place de programmes de différenciation cellulaire impliquant plusieurs centaines de gènes. Ces réponses peuvent être intégrées au niveau d'un ensemble d'individus partageant la même niche écologique. Les systèmes impliqués dans cette réponse adaptative sont les *systèmes intégrés*. Ils sont constitués d'un ensemble de protéines possédant généralement des fonctions biochimiques différentes et souvent portées par plusieurs domaines fonctionnels, et réalisant un réseau complexe d'interactions qui peuvent être stables ou transitoires. Des fonctions cellulaires complexes émergent des associations stables. Les systèmes appartenant à différentes espèces sont liés par des relations de parenté. Les gènes codant pour les différents partenaires de ces systèmes se situent en général dans une même proximité physique sur le chromosome. Parmi ces systèmes, nous distinguerons les transporteurs ABC. Ces systèmes sont de faible complexité, composés de moins d'une dizaine de partenaires avec des structures comprenant moins de cinq domaines. Leur identification n'est pas aisée car bien que possédant un nombre de partenaires restreint, il existe un grand nombre de systèmes dans les génomes et certains d'entre eux ont une faible conservation de séquence.
Les transporteurs ABC (figure 1.10), de **A**TP **B**inding **C**assette (cassette qui fixe l'ATP[19]),

[19] Adénosine Tri Phosphate

sont des systèmes d'export et d'import de molécules (le substrat) dans la cellule, présents à la fois chez les procaryotes et les eucaryotes (Higgins, 1992). Cependant, chez les eucaryotes, seuls les systèmes d'imports ont été identifiés (Taglicht et Michaelis, 1998) (Decottignies et Goffeau, 1997). Nous avons vu que la membrane plasmique agissait à la façon d'une barrière semi-perméable entre la cellule et l'environnement extra-cellulaire. Cette perméabilité doit toutefois être suffisamment sélective pour que des molécules essentielles comme le glucose, les acides aminés et les lipides puissent entrer facilement dans la cellule et y demeurer, et que des déchets du métabolisme puissent en sortir. En plus de leur capacité à être importeur ou exporteur, une caractéristique remarquable des transporteurs ABC est la grande variété de composés qu'ils peuvent transporter : il peut s'agir de petits solutés comme des ions mais également de molécules beaucoup plus grosses telles que des protéines (Fath et Kolter, 1993). Ces transporteurs sont parfois également impliqués dans le rôle de pathogénicité de certaines bactéries ou dans leur résistance aux antibiotiques. L'analyse du répertoire de tels systèmes peut donner de précieuses indications sur l'adaptation de l'organisme à son environnement (Holland et Blight, 1999) (Paulsen *et col.*, 1998). Un système de transport est typiquement composé de quatre régions fonctionnelles (ou domaines) ; chez les procaryotes ces domaines sont généralement portés par quatre gènes éventuellement organisés en opéron[20](Higgins, 1992) (chez les eucaryotes on peut retrouver ces domaines au sein d'une même protéine) :

- 2 domaines **MSD**(**M**embrane **S**panning **D**omain) qui constituent le pore par lequel le substrat traverse la membrane plasmique.
- 2 domaines **NBD**(**N**ucleotide **B**inding **D**omain) qui assurent le transport du substrat en fixant l'ATP et dont l'hydrolise permet de fournir de l'énergie par rupture d'une liaison covalente : ATP $\longrightarrow$ ADP + Pi.

Dans le cas des systèmes d'import, une protéine supplémentaire, appelée **SBP** (pour **S**olute **B**inding **P**rotein), est nécessaire : celle-ci se lie au substrat et permet son intégration dans le pore. De plus, c'est elle qui donne sa spécificité au système d'import (Tam et Saer, 1993).

1.5 Axes de recherche

L'analyse du répertoire des transporteurs ABC peut schématiquement se décomposer en cinq étapes :

1. Identification des partenaires
2. Assemblage
3. Reconstruction des systèmes incomplets
4. Classification en sous-familles
5. Identification du substrat

Les deux premières étapes sont réalisées lors de la création/mise à jour de la base de données ABCdb (Quentin et Fichant, 2000).

Un des problèmes lors de l'analyse des transporteurs ABC est la reconstruction des systèmes. Au cours de la diversification des transporteurs ABC, les gènes impliqués dans un même système

[20]Il n'est pas rare de trouver des transporteurs pour lesquels les gènes ne se situent pas tous sur le même brin.

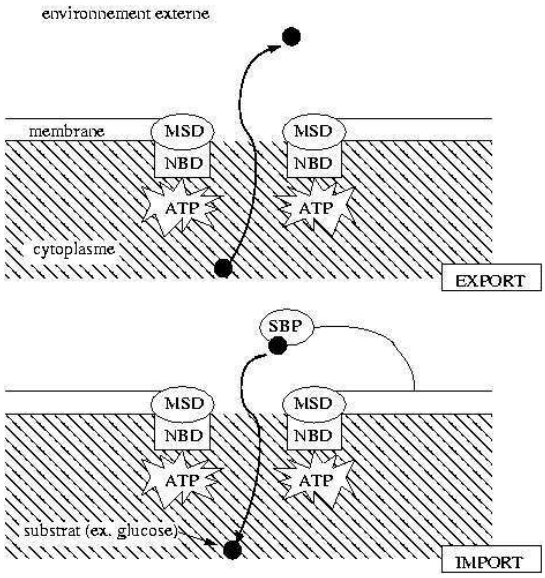


FIG. 1.10 – Schéma de deux transporteurs ABC : un exporteur et un importeur.

restent généralement groupés même si, parfois, des permutations et des inversions sont observées. La diversité peut également être acquise par la duplication partielle de systèmes (plusieurs protéines affines pour le même transporteur ou une ATPase pour énergiser plusieurs transporteurs). Pour l'étude de ces transporteurs, il est primordial de connaître les gènes impliqués dans chacun d'eux. Une méthode de reconstruction des transporteurs ABC a été mise au point (Quentin *et col.*, 1999), mais certains systèmes restent partiels si les gènes qui les composent sont disséminés sur le chromosome. Pour remédier à ce problème il a fallu élaborer une nouvelle stratégie, basée sur des comparaisons d'arbres de proximité (Quentin *et col.*, 2002). Or, cette méthode ne peut être automatisée, et ces informations sont primordiales pour l'annotation des génomes.

Un autre problème est la classification de ces systèmes. On les classe en sous-familles correspondant à de grandes classes de substrat tel que les ions, sucres, ... ((Tomii et Kanehisa, 1998), (Dassa *et col.*, 1993) et (Quentin *et col.*, 1999)). Il serait donc intéressant de classer ces systèmes en sous-familles plus précises.

Enfin, l'identification du substrat doit permettre d'identifier le rôle des transporteurs ABC étudiés. Elle sera élaborée sur le *principe du voisinage*, l'exploration des interactions entre les gènes pouvant se faire par l'analyse de l'organisation des génomes. Cette méthode peut se révéler puissante pour l'identification des rôles d'un gène dans la cellule (Nitschke *et col.*, 1998). La recherche de voisinages consiste alors à rassembler des objets proches à l'intérieur d'un même espace de caractéristiques. La proximité physique sur le chromosome est probablement la caractéristique la plus étudiée à cause de l'organisation de gènes au sein d'opérons mais les modules fonctionnels constituent un autre cas de voisinage intéressant : la fonction des gènes peut être suggérée par l'analyse des domaines de fusion de protéines dans les organismes où ces modules

constituent des gènes indépendants (Marcotte *et col.*, 1999). En étendant cette démarche, l'étude des systèmes ABC doit permettre d'identifier le *substrat* transporté : si des gènes extérieurs au système (enzymes ou familles de régulateurs par exemple) sont conservés dans de nombreux génomes, alors ces gènes peuvent aider à la prédiction du substrat – prédiction qui devra bien sûr être validée expérimentalement par la suite. L'hypothèse principale est que plus les gènes conservés seront loin du transporteur ABC, plus leur lien fonctionnel avec le transporteur sera ténu.

Ces cinq points permettent une meilleure caractérisation et annotation des systèmes de transport ABC. Je présenterai dans la suite les méthodes que j'ai développé pour répondre à ces questions.

2

Recherche de voisinages conservés : un cas de synténie locale

Tout ce qui existe dans l'Univers
est le fruit du hasard et de la nécessité.
Démocrite

DANS ce chapitre nous allons exploiter les relations de voisinages décrites dans le chapitre 1 (1.3.2) afin de compléter nos connaissances sur les transporteurs ABC. En effet, notre objectif est de développer une méthode permettant :

- d'identifier de nouveaux partenaires de ce système (protéines impliquées dans le passage d'un composé du périplasme à l'extérieur de la bactérie dans le cas des bactéries Gram-),
- de préciser la nature des composés transportés en identifiant les enzymes impliquées dans le métabolisme de cette molécule,
- d'identifier les gènes impliqués dans la régulation de l'expression des gènes codant pour les partenaires du système.

L'analyse des transporteurs ABC révèle que les gènes codant pour les différents partenaires d'un système ne sont pas systématiquement organisés en opéron. Ils peuvent être dans la même orientation ou dans des orientations différentes et interrompus par des gènes ne codant pas pour des protéines liées fonctionnellement au transporteur. Plus rarement ces gènes peuvent être dispersés sur le chromosome (cf Chapitre 4).

Pour répondre à ce problème, j'ai développé une méthode permettant de détecter des groupes de gènes conservés au voisinage d'un gène – dit gène d'ancrage – dans de nombreux génomes et ceci quelle que soit leur orientation, leur ordre ou leur proximité locale (les gènes conservés ne seront pas forcément côte à côte). D'après la théorie de l'évolution, nous savons que les protéines codées par des gènes issus d'un gène ancestral commun possèdent des fonctions identiques ou similaires.

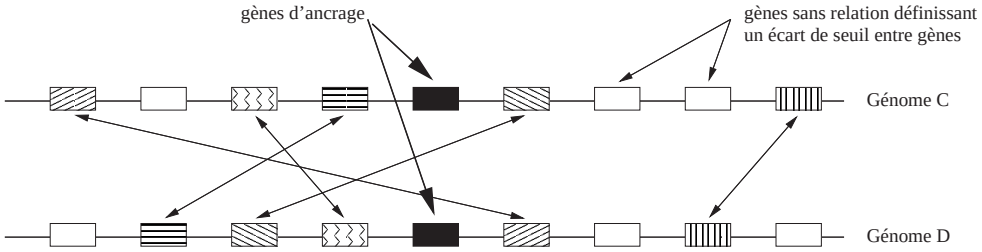


FIG. 2.11 – Représentation schématique du voisinage conservé autour de gènes d'ancrage (en noir) dans deux génomes C et D. Les gènes orthologues sont liés par une double flèche et possèdent le même motif. Les gènes en blanc sont des gènes sans relation.

C'est cette relation évolutive, appelée *orthologie*(Fitch, 1970), qui liera les gènes conservés entre génomes comme le montre la figure 2.11.
Les hypothèses principales de cette étude sont que plus les gènes conservés seront loin du gène d'ancrage, plus leur lien fonctionnel avec le gène d'ancrage sera ténu ; plus le nombre de gènes sans relation entre deux gènes conservés sera grand, plus la cohérence de l'ensemble des gènes conservés sera faible ; plus les génomes étudiés seront taxonomiquement distants, plus les conservations détectées seront fortes. Pour estimer le lien fonctionnel entre ces gènes, nous posons deux contraintes :

- Un gène ne sera considéré comme conservé que jusqu'à une certaine distance du gène d'ancrage.
- Dans le voisinage du gène d'ancrage, on ne tolèrera pas d'espacement entre gènes conservés supérieur à un seuil δ.

Dorénavant, lorsque nous parlerons d'*écart de seuil entre gènes*, il s'agira du nombre de gènes sans relation situés entre deux gènes possédant une relation génique dans la séquence comparée.
Cette problématique a été posée en 2000 alors qu'aucune méthode n'existait encore pour la résoudre. Historiquement trois méthodes ont été développées en parallèle : celles de Snel *et col.* (2000), Morgat et Viari (2001) et Colombo *et col.* (2001) (Colombo *et col.*, 2002). Puis, Bergeron *et col.* (2003) ont développé la méthode GENETEAMS. A des fins de clarté, j'utiliserai le formalisme introduit dans (Bergeron *et col.*, 2003) pour décrire la méthode que j'ai développé, méthode basée sur un formalisme dérivé des *CSPs* (**C**onstraints **S**atisfaction **P**roblems[21]) (Montanari, 1974). Par la suite, je présenterai l'outil de recherche d'instances récurrentes de gènes voisins STRING (Huynen et Snel, 2000), et l'algorithme GENETEAMS (Bergeron *et col.*, 2003). Enfin, je détaillerai les résultats obtenus en appliquant l'algorithme développé aux transporteurs ABC et je comparerai les trois méthodes.

2.1 Méthodologie

Bergeron *et col.* (2003), outre leur algorithme de recherche de groupes de synténie, proposent un formalisme permettant de clarifier la notion de distance intergénique qui peut être définie,

[21] Problèmes de Satisfaction de Contraintes

suivant les travaux, comme la différence de localisation physique des gènes sur le chromosome, la distance depuis une cible spécifique, ou bien le nombre de paires de bases. Je m'appuierai sur ce formalisme pour développer ma méthode dont le but est d'explorer le voisinage d'un gène donné – le gène d'ancrage – pour pouvoir inférer ses liens fonctionnels avec d'autres gènes. La recherche s'effectue en aval et en amont de la localisation chromosomique de ce gène. Nous prenons pour hypothèse que la conservation de gènes dans un même voisinage doit être le signe de liens fonctionnels entre leurs produits. Toutefois, nous relâchons la contrainte sur la distance intergénique, décomptée en nombre de gènes, où n'interviendra plus l'orientation des gènes. Dans un tel cadre, connaître la position d'un gène sur le chromosome constitue un élément fondamental puisque c'est à partir de cette position que nous pourrons déterminer l'écart de seuil entre gènes.

Définition 1.1 *Soit Σ un ensemble de n gènes appartenant au chromosome C, et $P_C : \Sigma \to \mathbb{Z}$ la fonction qui à chaque gène $g \in \Sigma$ associe un entier $P_C(g)$ qui sera sa position.*

Une fonction de ce type est très générale et permet de formaliser les différents types de distance intergénique. Elle induit une permutation sur un sous-ensemble S de Σ, ordonnant les gènes de S par position croissante.

Notation 1.2 *La permutation correspondant à l'ensemble des gènes Σ du chromosome C sera notée π_C*

Connaissant la position de deux gènes, nous définissons la distance qui les sépare.

Définition 1.3 *Soient g et g' deux gènes de Σ, la fonction $\Delta_C : \Sigma \times \Sigma \to \mathbb{Z}$ définit la distance entre ces deux gènes sur le chromosome C : $\Delta_C(g, g') = |P_C(g') - P_C(g)|$.*

Or dans notre cas, la distance entre un gène et le gène d'ancrage sera exprimée en nombre de gènes intercalés. Le gène d'ancrage occupe ici une position centrale par rapport à laquelle la position des autres gènes étudiés sera calculée. Nous devons donc étendre cette définition de la position.

Définition 1.4 *Soient deux gènes g et $\mathcal{A}$ du chromosome C. $\mathcal{A}$ est le gène d'ancrage. Alors la position du gène $g \in \Sigma$, exprimée par rapport à $\mathcal{A}$ sur le chromosome C, sera donnée par la fonction $P_{C_\mathcal{A}} : \Sigma \to \mathbb{Z}$ soit $P_{C_\mathcal{A}}(g) = P_C(g) - P_C(\mathcal{A})$.*

La position du gène d'ancrage est toujours 0, et les gènes situés en amont auront des positions négatives alors que les gènes situés en aval auront des positions positives. De plus, la distance séparant deux gènes est maintenant exprimée en nombre de gènes intercalés et elle est toujours calculée par rapport au gène d'ancrage. En nous basant sur la définition précédente :

Définition 1.5 *Soient g et $\mathcal{A}$ deux gènes de Σ où $g \neq \mathcal{A}$, la fonction $\Delta_{C_\mathcal{A}} : \Sigma \to \mathbb{N}$ définit la distance de g au gène d'ancrage sur le chromosome C, exprimée en nombre de gènes intercalés : $\Delta_{C_\mathcal{A}}(g) = |P_{C_\mathcal{A}}(g)| - 1$.*

Notre seconde hypothèse est que plus la distance entre gènes conservés[22] augmente, moins le lien fonctionnel est susceptible d'exister. Ainsi, un des paramètres essentiels de notre méthode est la taille maximale de l'écart de seuil entre gènes apparaissant entre un gène et le gène d'ancrage.

Notation 1.6 *Nous noterons δ la taille maximale de l'écart de seuil entre gènes (défini par l'utilisateur).*

Pour connaître la taille maximale de l'écart de seuil entre un gène et le gène d'ancrage, nous utiliserons une nouvelle fonction de distance :

Notation 1.7 *Soient g et $\mathcal{A}$ deux gènes du chromosome C où $\mathcal{A}$ est le gène d'ancrage ; soit $\mathcal{A}'$ le gène du chromosome D lié par une relation génique[23] à $\mathcal{A}$, $\Delta_{C_G} : \Sigma \rightarrow \mathbb{N}$ est la fonction indiquant le nombre maximum de gènes consécutifs entre g et $\mathcal{A}$ qui sont sans relation génique avec un quelconque gène du voisinage de $\mathcal{A}'$.*

Pour comparer la distribution des gènes entre deux fragments chromosomiques, nous devons être en mesure de dire si le gène g du chromosome C est le gène orthologue de g' sur le chromosome D (et réciproquement). Notons ici que la relation génique choisie peut être différente de l'orthologie mais doit être symétrique – ou bijective. Dans le cas d'une relation non bijective, un gène g du chromosome C pourrait être lié à g' du chromosome D, lui-même lié à g'' du chromosome C (figure 2.12). Nous ne saurions alors sur quel gène (g ou g'') porte la relation de g'. Avec ce genre de relation, une solution est de considérer que g et g'' représentent un seul et même gène qui a été scindé en deux parties au cours de l'évolution.

Notation 1.8 *Soient g et g' deux gènes des chromosomes C et D respectivement, $p_{CD}(g, g')$ est une paire de gènes liés par une relation génique bijective. Utilisant l'orthologie, pour simplifier les notations, nous dirons que :*

$$p_{CD} : \Sigma \times \Sigma \rightarrow \Sigma \cup \{\emptyset\} \text{ est telle que : } p_{CD}(g, g') = \begin{cases} g \text{ si } g \text{ et } g' \text{ sont orthologues} \\ \emptyset \text{ sinon} \end{cases}$$

Par cette fonction, deux gènes g et g' liés par une relation génique bijective auront donc la même dénomination ; ils seront identifiés par le même nom sur les différents chromosomes. Pour la description de la méthode, nous considèrerons que les comparaisons sont effectuées entre fragments chromosomiques et que l'un d'entre eux est pris comme référence pour être comparé à tous les autres (la comparaison s'effectuant deux à deux). Le problème peut alors être exprimé comme suit : soit $\mathcal{A}_C$ un gène d'ancrage issu du chromosome de référence C, et $\mathcal{A}_D$ son orthologue sur le chromosome D ; notre objectif est de trouver la liste des gènes liés $p_{CD}(g, g')$ conservés dans le voisinage de la paire de gènes d'ancrage $p_{CD}(\mathcal{A}_C, \mathcal{A}_D)$ avec un écart de seuil entre gènes tel que $\Delta_{C_G}(g) \leq \delta$ et $\Delta_{D_G}(g') \leq \delta$.

[22] Les gènes conservés sont ceux qui sont communs autour de $\mathcal{A}$ dans plusieurs génomes.
[23] Relation pouvant être modifiée suivant les besoins.

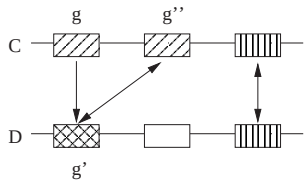


FIG. 2.12 – Illustration d'une relation non bijective sur deux chromosomes C et D. Les gènes g et g'' doivent être considérés comme un seul gène scindé au cours de l'évolution et en relation avec le gène g'.

Propriété 1.9 *La liste des gènes conservés à un seuil $\delta - 1$ est incluse dans la liste des gènes conservés à un seuil δ.*

Pour une recherche exhaustive à tous les seuils inférieurs à δ, nous pourrons donc commencer par rechercher la liste des gènes conservés au seuil δ puis en déduire les listes des gènes conservés à des seuils inférieurs. Comme nous étudions le voisinage d'un gène particulier, nous devons définir ce voisinage pour nos calculs.

Notation 1.10 *On appellera fenêtre de taille w, le nombre de gènes étudiés en aval et en amont du gène d'ancrage (soit $2w + 1$ gènes au total).*

J'ai commencé par développer un algorithme très simple, basé sur des intersections successives de listes de gènes.

2.1.1 Une première approche par intersections de listes

En phase initiale, deux listes contiennent l'ensemble des gènes g_i et g'_i appartenant respectivement à deux génomes différents C et D :

- qui sont à une distance du gène d'ancrage inférieure à la taille de la fenêtre w : $\Delta_{C_\mathcal{A}}(g) \leq w$ et $\Delta_{D_\mathcal{A}}(g') \leq w$,
- et qui appartiennent à des paires de gènes liés par une relation génique : $\exists g$ tel que $p_{CD}(g, g') \neq \emptyset$ et $\exists g'$ tel que $p_{CD}(g, g') \neq \emptyset$.

Ensuite, tant que l'écart de seuil maximal entre gènes est supérieur à δ, les gènes situés entre le dernier gène délimitant cet écart et la fin de la séquence sont supprimés de la liste (gène bordure de l'écart compris). Ce processus de suppression d'un (ou plusieurs) élément(s) d'une liste implique un processus d'intersection entre les deux listes pour propager les modifications effectuées sur une liste à la seconde liste. Or, le fait de supprimer un gène peut augmenter la taille d'un écart de seuil entre gènes et un processus de suppression des éléments trop éloignés devra être réappliqué. Ainsi, les procédures de suppression et d'intersection sont appliquées jusqu'à la convergence de la méthode. A la fin, on obtient donc deux listes, contenant un sous-ensemble des gènes initiaux, qui sont les gènes conservés pour une fenêtre fixée de taille w et un seuil δ.
Lors du calcul des voisinages conservés à tous les seuils possibles, on part du seuil maximal

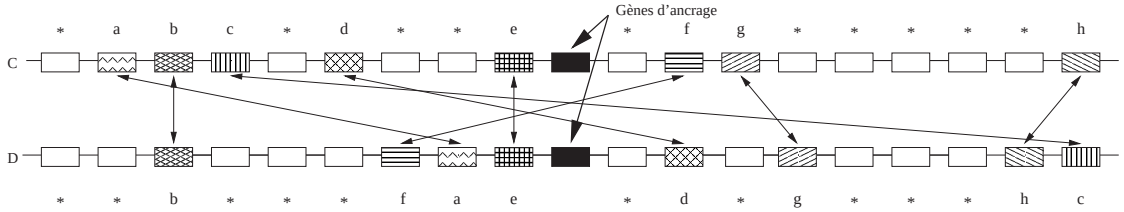


FIG. 2.13 – Conservation entre deux séquences de chromosomes C et D. Le "nom" de chaque gène est porté au-dessus de ce dernier dans le chromosome C (respectivement au-dessous dans le chromosome D). Le nom des gènes sur le chromosome D a été recodé à l'aide de la fonction p_{CD}. Ainsi, pour le gène a sur le chromosome C qui était en relation avec le gène x sur D, $p_{CD}(a, x) = a$ et le gène x a été renommé en a. Pour une meilleure lisibilité, les 0 renvoyés par la fonction en cas d'absence de relation entre gènes ont été remplacés par des "*".

$\delta = w$ et on utilise la liste calculée pour $\delta = t$ pour inférer la liste des gènes conservés à $t - 1$ et ce jusqu'à $\delta = 0$. Le fait de ne pas fixer la valeur du δ permet de ne perdre aucune information : on a ainsi des gènes conservés avec une forte confiance – correspondant à un δ faible – et des gènes conservés avec une confiance moindre – correspondant à un δ élevé.
Sur l'exemple présenté en figure 2.13, on étudie deux séquences chromosomiques en prenant une fenêtre de $w = 9$ gènes de part et d'autre des gènes d'ancrage. Des deux séquences, nous obtenons les listes :

$$\begin{array}{lcccccccccccccccccc} L_C & = & * & a & b & c & * & d & * & * & e & \mathcal{A} & * & f & g & * & * & * & * & * & h \\ L_D & = & * & * & b & * & * & * & f & a & e & \mathcal{A} & * & d & * & g & * & * & * & h & c \end{array}$$

$\mathcal{A}$ désignant le gène d'ancrage, l'écart de seuil entre gènes le plus important est de 5 gènes ; donc pour un δ variant de $w = 9$ à 5, tous les gènes seront conservés.
En considérant un $\delta = 4$: suppression du gène h dans L_C car trop éloigné en terme d'écart entre gènes, puis intersection avec L_D.

$$\begin{array}{lcccccccccccccccccc} L_C & = & & a & b & c & * & d & * & * & e & \mathcal{A} & * & f & g & & & & & & \\ L_D & = & & & b & * & * & * & f & a & e & \mathcal{A} & * & d & * & g & * & * & * & * & c \end{array}$$

Pour un $\delta = 3$: suppression du gène c dans L_D et intersection avec L_C.

$$\begin{array}{lcccccccccccccccccc} L_C & = & & & a & b & * & * & d & * & * & e & \mathcal{A} & * & f & g & & & & & \\ L_D & = & & & & b & * & * & * & f & a & e & \mathcal{A} & * & d & * & g & & & & \end{array}$$

Pour un $\delta = 2$: suppression du gène b dans L_D puis intersection avec L_C. Or le fait de supprimer b dans L_C va augmenter l'écart de seuil entre les gènes a et d. Cette distance est portée à 3 et dépasse le seuil δ. Il faut donc supprimer le gène a dans L_C et effectuer l'intersection avec L_D.

$$\begin{array}{lcccccccccccccccccc} L_C & = & & & & & & d & * & * & e & \mathcal{A} & * & f & g & & & & & & \\ L_D & = & & & & & & & f & * & e & \mathcal{A} & * & d & * & g & & & & & \end{array}$$

Et ainsi de suite jusqu'à $\delta = 0$.
Cet algorithme est polynomial en temps. En posant n le nombre de gènes étudiés, pour un seuil

δ donné, la phase d'initialisation (création des listes) est réalisée en n opérations et l'intersection entre deux listes de gènes peut être effectuée en temps linéaire. Mais dans le pire des cas, à chaque étape de l'algorithme, un seul gène sera supprimé (alternativement sur chaque génome) et cette modification devra être propagée en effectuant une nouvelle intersection. Il faudra donc effectuer n intersections : la complexité globale de cet algorithme est donc en $O(n^2)$.
Pour explorer des génomes complets il fallait un algorithme plus performant. Nous sommes en présence de contraintes, je me suis donc tourné vers les CSPs. En effet, ce formalisme permet de représenter des problèmes basés sur des contraintes. Cette approche, à la fois originale et élégante, est également beaucoup plus performante dans le cas particulier qui nous intéresse.

2.1.2 Les Problèmes de Satisfaction de Contraintes

Le formalisme CSP a été introduit par (Montanari, 1974). Il permet d'exprimer une multitude de problèmes de nature totalement différente. En bioinformatique, les CSP et techniques d'Intelligence Artificielle ont été utilisés notamment dans (Gaspin *et col.*, 1995). Mais, plus classiquement, il s'agit de problèmes de coloration de graphes, de logique propositionnelle, d'ordonnancement, etc. Dans tous ces problèmes, il est possible d'exprimer sous forme de contraintes les propriétés et les relations qui existent entre les objets manipulés. De plus, ces contraintes peuvent être décrites de multiples façons : par une équation, une inéquation, un prédicat, une fonction booléenne, etc. Je m'attacherai ici à rappeler brièvement le formalisme introduit par Montanari (Montanari, 1974), puis, je présenterai l'extension des CSPs aux problèmes temporels proposée par Dechter *et col.* (1991). Enfin, je montrerai comment ces TCSPs (pour **T**emporal **C**onstraint **S**atisfaction **P**roblems[24]) peuvent être adaptés à des problèmes spatiaux.
Un CSP se définit par la donnée d'un ensemble de variables et d'un ensemble de contraintes. Chaque variable peut prendre une valeur choisie dans le domaine qui lui est associé. Les contraintes, quant à elles, décrivent les combinaisons autorisées de valeurs pour les variables. Pour résoudre le problème, on attribue alors une valeur à chaque variable de sorte que toutes les contraintes soient satisfaites.

Définition 1.11 *Une instance CSP est un quadruplet (X, D, C, R) où :*
- *X est un ensemble $\{x_1, x_2, ..., x_n\}$ de variables.*
- *D est un ensemble de domaines finis $\{d_1, d_2, ..., d_n\}$ tel que le domaine d_i soit associé à la variable x_i de X. Le domaine d_i contient les valeurs que peut prendre la variable x_i.*
- *C est un ensemble $\{c_1, c_2, ..., c_m\}$ de contraintes. Chaque contrainte c_i de C est définie par l'ensemble de variables sur lesquelles elle porte.*
- *R est un ensemble $\{r_1, r_2, ..., r_m\}$ de relations tel que la relation r_i soit associée à la contrainte c_i. La relation r_i est définie sur $\prod_{x \in c_i} d_x$. Elle représente les affectations compatibles entre les variables contraintes par c_i.*

Afin de représenter la structure du problème, on emploie un hypergraphe ou graphe de contraintes.

[24] Problèmes de Satisfaction de Contraintes Temporels

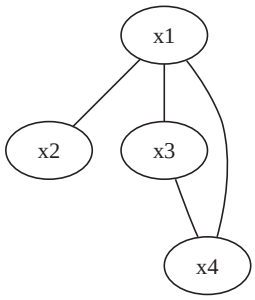


FIG. 2.14 – Graphe de contraintes associé au CSP $\mathcal{P}$.

Définition 1.12 *Etant donnée une instance CSP $\mathcal{P} = (X, D, C, R)$, l'hypergraphe (X, C) est l'hypergraphe de contraintes associé au problème $\mathcal{P}$. Si le CSP $\mathcal{P}$ est binaire*[25] *il s'agit d'un graphe, dit graphe de contraintes.*

On désigne également le graphe de contraintes sous le terme *réseau de contraintes*. Voici un exemple très simple de CSP accompagné de son réseau de contraintes en figure 2.14 :
Considérons le CSP binaire $\mathcal{P} = (X, D, C, R)$ avec :

- $X = \{x_1, x_2, x_3, x_4\}$,
- $D = \{d_1, d_2, d_3, d_4\}$ avec $d_1 = d_2 = d_3 = d_4 = \{1, 2, 3\}$,
- $C = \{c_{12}, c_{13}, c_{14}, c_{34}\}$ avec $c_{ij} = \{x_i, x_j\}$,
- $R = \{r_{12}, r_{13}, r_{14}, r_{34}\}$.

La définition des relations est la suivante :

- $r_{12} : x_1 \neq x_2$,
- $r_{13} : x_1 \leq x_3$,
- $r_{14} : x_1 > x_4$,
- $r_{34} : x_3 \neq x_4$.

Pour résoudre un tel problème, il faut trouver les affectations de variables vérifiant les contraintes données. On parle d'une *instanciation* de variables.

Définition 1.13 *Etant donné $Y \subseteq X$ avec $Y = \{y_1, ..., y_k\}$, une instanciation des variables de Y est un k-uplet $(v_1, ..., v_k)$ de $d_1 \times ... \times d_k$ où d_i est le domaine associé à la variable y_i. Une instanciation est dite complète si elle porte sur toutes les variables de X, et partielle sinon.*

Dans la littérature, on pourra trouver le terme d'*affectation* au lieu d'*instanciation*.

Notation 1.14 *Soit A une instanciation, nous noterons X_A l'ensemble des variables affectées dans A. Etant donné un ensemble $Y \subseteq X$, $A[Y]$ correspond à l'affectation A restreinte aux variables qui sont à la fois dans Y et dans X_A.*

[25] On parle de CSP binaire lorsque les contraintes qui le composent portent sur deux variables.

Définition 1.15 *Une instanciation A satisfait une contrainte c de C si $A[c] \in r_c$, sinon A viole c (considérant que les variables de A contiennent les variables de la contrainte). Une instanciation A est cohérente si $\forall c \in C, c \subseteq X_A, A[c] \in r_c$. Elle est dite incohérente sinon.*

Souvent, le terme de *consistant* est employé en lieu et place de *cohérent*. Une solution du CSP est donc une *instanciation complète consistante*. Sur l'exemple précédent, $\{x_1 \leftarrow 2, x_2 \leftarrow 3, x_3 \leftarrow 3, x_4 \leftarrow 1\}$[26] est une solution. Le problème de détermination de l'existence d'une solution est NP-complet.

Pour définir des problèmes temporels, de nouveaux formalismes, dérivés des CSPs, ont été proposés : les TCSPs. Il existe deux types de raisonnements temporel : le raisonnement qualitatif (Allen, 1983), et le raisonnement quantitatif (Dechter *et col.*, 1991). Dans le modèle qualitatif, les variables peuvent représenter des points du temps ou des intervalles et les contraintes décrivent l'ordre des variables. Dans le modèle quantitatif, les variables représentent des points du temps et les contraintes sont exprimées comme des distances entre variables. Il paraît évident que le second type de raisonnement est le plus à même de représenter le problème posé. Je ne développerai donc ici que ce raisonnement, utile à mon propos. Un TCSP est un CSP dans lequel :

- Les variables sont des points de $\mathbb{R}$. Une variable représente soit le début, soit la fin d'un événement,
- Le domaine de chaque variable est la droite des réels,
- Les contraintes expriment l'information temporelle qui existe entre les événements (distance entre les événements). Elles peuvent être :
  - unaires : pour une variable x_i, x_i appartient à une liste d'intervalles $\{[a_1, b_1], ...[a_n, b_n]\}$, alors la contrainte sera $c_i : (a_1 \leq x_i \leq b_1) \vee .. \vee (a_n \leq x_i \leq b_n)$
  - binaires : pour deux variables x_i, et x_j, la distance $x_j - x_i$ appartient à une liste d'intervalles $\{[a_1, b_1], ..., [a_n, b_n]\}$, alors la contrainte sera $c_i : (a_1 \leq x_j - x_i \leq b_1) \vee .. \vee (a_n \leq x_j - x_i \leq b_n)$.

Tout comme pour les CSPs, on peut représenter un TCSP par un graphe de contraintes. Il s'agit d'un graphe orienté dans lequel les sommets représentent l'ensemble des variables, et les arcs représentent les contraintes. Ils sont étiquetés par la liste des intervalles contraignant les deux sommets qu'ils lient. Ainsi, une arête (a, b) étiquetée par $[x, y]$ indique-t-elle que pour aller du sommet a au sommet b, il faudra un temps compris entre x et y.
On peut bien sûr avoir des problèmes dans lesquels les contraintes sont plus simples.

Définition 1.16 *Un problème temporel simple ou STP (pour **S**imple **T**emporal **P**roblem) est un TCSP dans lequel chaque contrainte est réduite à un seul intervalle.*

Ainsi, dans un STP, les contraintes ne sont exprimées que par deux inéquations. Un STP peut être représenté par un graphe de distance.

[26] La notation formelle (mais de lecture moins aisée) de cette solution est $(2, 3, 3, 1)$.

Définition 1.17 *A chaque STP $\mathcal{P}$ on peut associer un graphe de distance G_d (encore appelé d-graphe) qui est un graphe orienté dont les arcs sont affectés d'un poids. G_d a le même ensemble de sommets que le graphe des contraintes de $\mathcal{P}$, et chaque arc (i, j) est affecté d'un poids a_{ij} qui représente $x_j - x_i \leq a_{ij}$.*

Les STPs sont des problèmes dont la complexité maximale en temps est en $O(n^3)$ (Dechter *et col.*, 1991). Dans le cas de notre problème, nous allons nous placer dans un cas plus particulier de STP.

2.1.3 Algorithme

L'algorithme que j'ai développé s'inspire du modèle quantitatif, proposé par le formalisme STP, et qui peut être transposé aux données spatiales sur une dimension, tels que des gènes ordonnés le long d'un chromosome. En effet, on représente des points sur une droite : que cette droite représente une échelle temporelle ou métrique, les informations n'en seront pas modifiées. Dans un tel modèle, les variables seront les différents gènes présents dans la fenêtre définie par la taille w et centrée sur le gène d'ancrage $\mathcal{A}$. Nous aurons ici deux types de contraintes qui seront exprimées par des équations linéaires (d'où une simplification du problème par rapport aux inéquations des STPs traditionnels) :

- C_{d_X} : contrainte de distance exprimant la distance entre un gène et le gène d'ancrage sur le chromosome X. Il s'agit de la distance $\Delta_{X\mathcal{A}}$, définie en p. 31, à la différence près que nous voulons conserver l'information sur la position du gène sur le chromosome par rapport au gène d'ancrage. Cette position $P_{X_\mathcal{A}}(g)$ nous permet de définir une contrainte où la distance sera négative si le gène g se situe en aval du gène d'ancrage et positive sinon :

$$C_{d_X} : g - \mathcal{A} = \begin{cases} \Delta_{X\mathcal{A}}(g) \text{ si } P_{X_\mathcal{A}} \geq 0 \\ -\Delta_{X\mathcal{A}}(g) \text{ sinon} \end{cases}$$

  Cette contrainte ne sera calculée qu'une seule fois, la position des gènes sur le chromosome étant fixe.
- C_{g_X} : contrainte d'écart de seuil entre gènes exprimant la taille du plus grand écart existant entre un gène et le gène d'ancrage sur le chromosome X. Il s'agit de la distance Δ_{XG} définie en p. 32 :

$$C_{g_X} : g - \mathcal{A} = \Delta_{XG}(g)$$

  Cette contrainte est susceptible d'être recalculée au cours du processus de résolution, la suppression d'un gène pouvant augmenter la taille d'un écart de seuil entre gènes.

Il existe également une contrainte sous-entendue qui est que lors de la recherche de la conservation à un seuil δ, on doit avoir $C_{g_X} \leq \delta$.
Le déroulement de l'algorithme sera alors le suivant : dans une première phase, dite *phase de révision*, on supprime tous les gènes qui violent la contrainte $C_{g_X} \leq \delta$. Pour chacun de ces gènes, si leur suppression implique un accroissement de l'écart de seuil entre gènes et oblige d'autres gènes à violer la contrainte, alors on *propage* l'information à ces gènes. Ce processus de révision/propagation est appliqué jusqu'à la convergence. Pour un calcul de la conservation à tous les seuils possibles, comme précédemment, nous utiliserons la propriété de la p. 33 : "*(...)*

la liste des gènes conservés à un seuil $\delta - 1$ est incluse dans la liste des gènes conservés à un seuil δ.". Appliquons maintenant cet algorithme à l'exemple de la figure 2.13 (rappelé en figure 2.15).

Notation 1.18 *A chaque paire de gènes (g, g') des chromosomes C et D, telle que $p_{CD}(g, g') \neq \{\emptyset\}$, nous associons un quadruplet $\{C_{d_C}(g), C_{d_D}(g'), C_{g_C}(g), C_{g_D}(g')\}$.*

Le problème peut être représenté sous la forme d'un *d-graphe* où les arcs de relation entre gènes (où aucun des deux n'est le gène d'ancrage) peuvent être inférés des contraintes C_{d_X} (figure 2.15). Ces arcs, qui ne font pas partie du d-graphe, indiquent la zone d'influence de chaque gène. Ainsi, une modification sur le gène g peut modifier les gènes h et c sur le chromosome D et c sur le chromosome C.
Si l'on recherche l'ensemble des gènes conservés dans le voisinage de $\mathcal{A}$ (où la fenêtre est de $w = 9$ gènes de part et d'autre de $\mathcal{A}$) pour un seuil $\delta = 2$, nous obtiendrons :

- Suppression des gènes (phase de révision) :
  - b car $C_{g_D}(b) = 3 > \delta$
  - c car $C_{g_D}(c) = 3 > \delta$
  - h car $C_{g_C}(h) = 5 > \delta$ et $C_{g_D}(h) = 3 > \delta$
- Phase de propagation : la suppression des gènes b et c sur le chromosome C entraînent un accroissement de l'écart de seuil entre gènes. De plus, b et c se situaient entre le gène a et le gène d'ancrage $\mathcal{A}$. Donc $C_{g_C}(a)$ est mis à jour.
- Phase de révision : la nouvelle contrainte d'écart de seuil entre gènes sur a viole $C_{g_C}(a) \leq \delta$, donc le gène a est supprimé.

Ainsi, pour un seuil $\delta = 2$, les gènes conservés dans le voisinage de $\mathcal{A}$ sont $\{d, e, f, g\}$.
Pour des séquences chromosomiques de n gènes, durant la phase d'initialisation, les contraintes C_{d_X} et C_{g_X} sont calculées avec une complexité de l'ordre de $O(n)$. En considérant que les n gènes sont conservés à chaque itération, les phases de révision et de propagation sont effectuées en n étapes. En recherchant les voisinages conservés à tous les seuils possibles, il y a au plus $\delta_{max} = w$ itérations et les opérations précédentes sont donc effectuées en $\delta_{max} \times n$. Toutefois, dans le pire des cas, la complexité de notre algorithme est en $O(n)$. En effet, en pratique n décroît rapidement avec δ.

Destiné à une utilisation sur des génomes bactériens, le traitement des génomes circulaires a été pris en compte. Avec cet algorithme il ne s'agit en fait que d'une légère modification de la fonction $P_{X_\mathcal{A}}$ donnant la position d'un gène. On considère simplement que le gène qui précède le premier gène sur le chromosome considéré est en fait le dernier gène (et réciproquement). Pour ce qui est de la recherche de voisinage sur des génomes multiples, notre algorithme effectuant des comparaisons deux à deux en prenant pour référence les gènes de la première séquence chromosomique, nous effectuons les recherches de voisinage en considérant tour à tour chaque génome comme génome de référence. Une généralisation de l'algorithme pour une comparaison multiple (sans passer par les comparaisons deux à deux) est possible mais entraînerait une perte d'information : les conservations de gènes devraient être observées dans absolument toutes les séquences pour être détectées. Or ici, nous pouvons détecter des gènes qui, par exemple, seraient conservés entre espèces très proches mais seraient conservés très rarement dans les autres espèces.

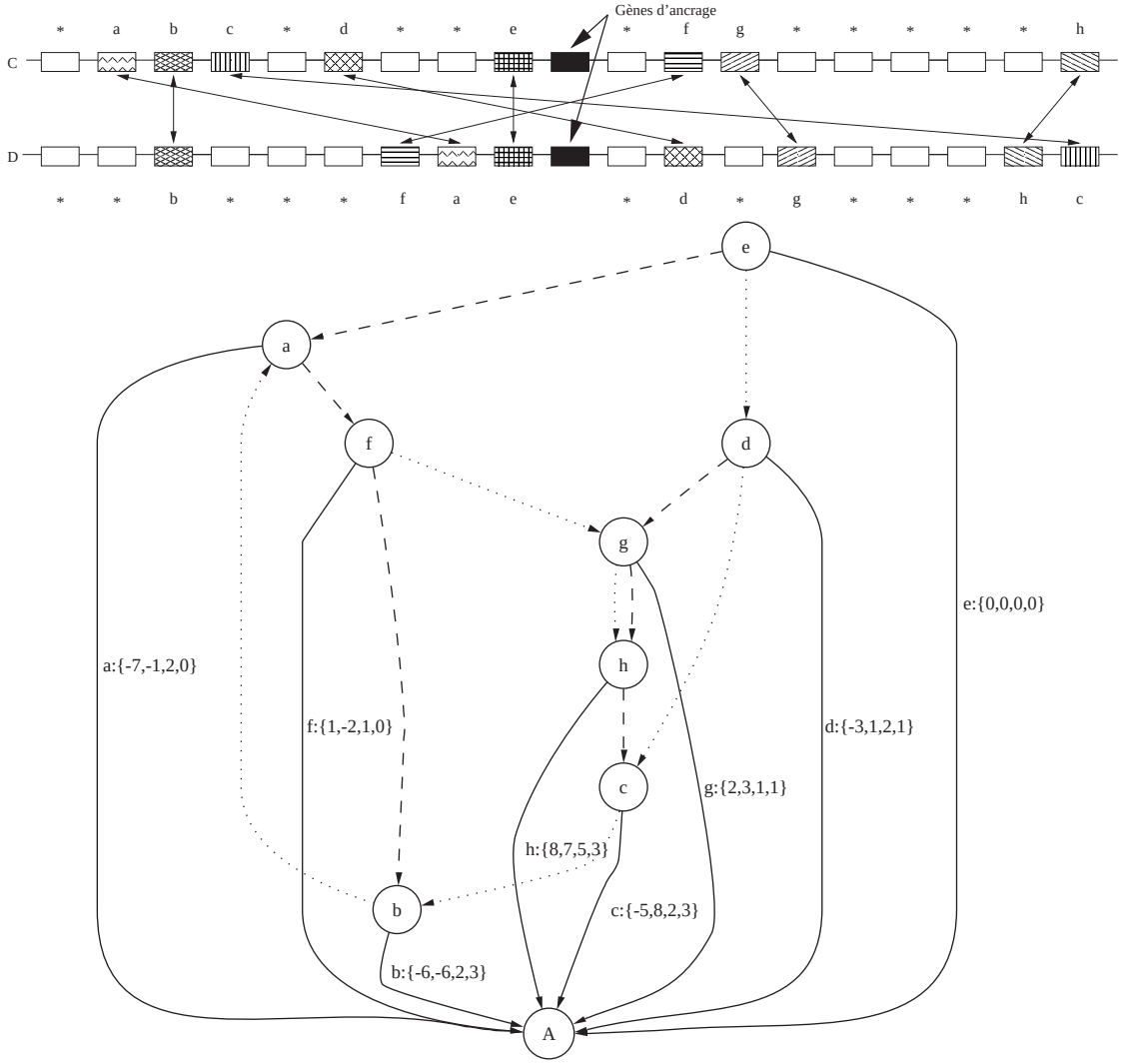


FIG. 2.15 – *d-graphe* de l'exemple 2.13. Les arcs notés en pointillés sont les arcs inférés des contraintes C_{d_X} : en pointillés courts, ceux issus du chromosome C et en pointillés longs, ceux issus du chromosome D. Ainsi, pour le gène a, les pointillés longs indiquent que sur le chromosome D, ce gène se trouve entre les gènes e et f. Les pointillés courts indiquent eux que sur C, le gène a se trouve à une extrémité du fragment chromosomique et qu'il est précédé par le gène b. Les informations portées par l'arc plein $\{-7,-1,2,0\}$ indiquent que $C_{d_C}(a) = -7$ et $C_{d_D}(a) = -1$: a se trouve à une distance de 7 gènes en aval du gène d'ancrage $\mathcal{A}_C$ sur C et à un gène en aval du gène d'ancrage $\mathcal{A}_D$ sur D ; $C_{g_C}(a) = 2$ et $C_{g_D}(a) = 0$: l'écart de seuil entre les gènes a et $\mathcal{A}_C$ vaut 2 (il y a deux gènes intercalés sans relation génique dans les fragments chromosomiques étudiés), et entre a et $\mathcal{A}_D$ elle vaut 0 (aucun gène intercalé).

2.2 Les méthodes développées en parallèle

Les deux travaux que je m'apprête à présenter, (Snel *et col.*, 2000) et (Bergeron *et col.*, 2003) sont ceux qui se rapprochent le plus de ma méthode. De nombreuses autres méthodes, recherchant la conservation d'un certain ordre à l'intérieur des génomes, ont été développées. Elles ont pour but de prédire la fonction des gènes et utilisent des relations d'homologie et de voisinage. De plus, la masse d'informations traitée est telle qu'il faut rechercher des algorithmes très performants pour traiter le problème en un temps acceptable. Je citerai rapidement (Mazumder *et col.*, 2001), (Tamames *et col.*, 2001), et (Suyama et Bork, 2001) dont l'objectif est surtout de montrer que la conservation des gènes n'est pas aléatatoire. Abordons maintenant la plus aboutie de ces méthodes, puisque dotée d'une interface conviviale et accessible directement sur internet : le serveur web STRING.

2.2.1 STRING

La méthodologie de STRING a été décrite dans (Snel *et col.*, 2000) et une mise à jour a été faite (von Mering *et col.*, 2003), portant sur l'augmentation du nombre de génomes disponibles ainsi que sur l'amélioration de l'interface web et des outils d'analyse exploratoire disponibles (*http ://www.bork.embl-heidelberg.de/STRING*). L'utilisateur doit fournir un gène de requête qui sera utilisé comme gène d'ancrage – le *seed gene*. S'il n'y a aucun gène conservé dans le voisinage de ce gène, alors ce sont ses gènes orthologues qui seront utilisés en tant que gène d'origine (ancrage). Le processus est effectué par itérations successives. Dans la première itération, STRING récupère et affiche les gènes qui apparaissent de manière répétée en co-occurrence avec le gène d'origine dans des groupes de gènes de multiples génomes issus de la bangue de données *SwissProt*. Les groupes de gènes sont ici définis en utilisant le concept de gènes en *série* d'Overbeek *et col.* (1999).

Définition 2.1 *Un ensemble de gènes en série – ou run – est un ensemble de gènes sur le même brin qui ne sont pas interrompus par des séquences de plus de* 300 *paires de bases ne codant pour aucun gène (Overbeek* et col.*, 1999).*

Il faut noter que deux gènes ayant fusionné au cours de l'évolution seront définis comme appartenant à une même *série* ; le gène résultant portant alors deux domaines fonctionnels. Dans les itérations suivantes, ce processus sera répété en utilisant successivement tous les nouveaux gènes, découverts lors de l'itération précédente, comme gène d'origine. Le nombre d'itérations est fixé par l'utilisateur ; dans la dernière version ce paramètre est masqué par un autre : le nombre maximum de gènes en interaction (la valeur par défaut est fixée à 10). Le processus général s'achève lorsque ce nombre est atteint ou lorsqu'aucun nouveau gène n'est découvert (convergence).
Dans la dernière version de STRING, les orthologues sont issus de la base de données COG (pour **C**lusters of **O**rthologous **G**enes) (Tatusov *et col.*, 2001). Comme cette base (voir Chapitre 1) n'est pas mise à jour aussi rapidement que les génomes nouvellement séquencés apparaissent,

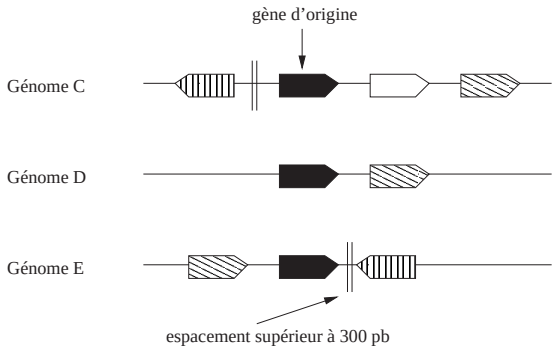


FIG. 2.16 – Fonctionnement de STRING : voisinage conservé autour de gènes d'origine (en noir) dans trois génomes C, D et E. Les gènes orthologues possèdent le même motif, les gènes en blanc sont des gènes ne possédant pas d'orthologue, et les doubles barres indiquent la fin d'une *série*. La pointe se situant à l'avant d'un gène indique le sens de transcription du gène (ou le brin auquel il appartient). Il s'agit ici de la première itération. Dans la seconde itération, les gènes hachurés seront tour à tour considérés comme gène d'origine. Puis les itérations se succèdent jusqu'au nombre fixé ou jusqu'à la convergence.

elle est enrichie par des prédictions : par analyse de similarité les nouveaux gènes sont affectés à un groupe COG. Les groupes de gènes découverts sont ensuite affichés graphiquement, accompagnés d'une table indiquant le nombre de fois où le gène d'origine apparaît en co-occurrence avec chaque autre gène dans la même *série*. Ceci permet d'apprécier le degré de l'association génomique entre ces deux gènes, et donc d'évaluer la force de l'association fonctionnelle de leurs produits. Un exemple de déroulement de cet algorithme est donné en figure 2.16.
Cette méthode est simple, efficace et rapide. En outre, elle bénéficie dans sa version révisée d'une très bonne interface graphique aux informations multiples. De plus, des outils de prédictions supplémentaires sont utilisés en parallèle : fouille de données dans les textes scientifiques, données expérimentales, ... Tout ceci en fait donc un outil de choix pour l'étude de la conservation du voisinage d'un gène. Toutefois, nous ne pouvons pas ignorer certaines limites du système :

- L'utilisateur doit fixer le nombre maximum de gènes en interaction. La méthode utilise des gènes d'ancrage différents au fil des itérations et ces gènes appartiennent à des groupes COG qui peuvent contenir de nombreux éléments. Il y a donc un risque d'explosion : la masse d'informations sera trop grande pour être exploitable. C'est notamment ce qui se produit avec certains gènes de transporteurs ABC car cette famille contient des gènes hautement paralogues.
- A l'intérieur des groupes de gènes conservés, les gènes doivent tous avoir la même orientation (ce fait provenant de l'utilisation de la notion de gènes en *série*). Or, il existe des groupes de gènes fonctionnellement liés qui présentent la particularité de contenir indifféremment des gènes sur un brin ou sur l'autre (en sens codant ou en sens inverse). C'est le cas de nombreux systèmes de transport ABC. STRING ne permet donc que d'obtenir des résultats très partiels pour de tels systèmes.

2.2.2 GeneTeams

Le formalisme de cette méthode ayant déjà été employé, je renverrai le lecteur en page 31 pour les définitions de la position d'un gène, la permutation et la distance. Ces notions nous permettent de définir ce que les auteurs appellent des δ-chaînes. Il s'agit de groupes de gènes d'orientation quelconque dans lesquels la distance entre deux gènes consécutifs n'est pas plus grande que le seuil δ.

Définition 2.2 *Soit S un sous-ensemble de Σ du chromosome C, et $(g_1...g_k)$ la permutation induite sur S. Pour $\delta > 0$, l'ensemble S est une δ-chaîne du chromosome C si pour $1 \leq j < k$, $\Delta_C(g_j, g_{j+1}) \leq \delta$.*

Ainsi, en prenant pour exemple un chromosome D avec $\Sigma = \{a, b, c, d, e, f, g\}$, où l'on note par une étoile les gènes qui ne sont pas identifiés dans Σ, posons $D = c\ a * e * d * * * b\ g\ f$.
Alors, si $\delta = 2$, $\{a, e\}$, $\{e, d\}$, $\{a, e, d\}$ sont des δ-chaînes, mais $\{a, d\}$ ne l'est pas (si l'on ne tient pas compte du e entre a et d alors $\Delta_D(a, d) = 4$). On peut noter que tous les singletons sont des δ-chaînes.

Définition 2.3 *Une δ-chaîne maximale sur un chromosome C est un ensemble de δ-chaînes $\{d_1...d_k\}$ telles que pour $1 \leq j < k$, $\Delta_C(d_j^d, d_{j+1}^p) > \delta$ où d_j^d est le dernier élément de la δ-chaîne d_j et d_{j+1}^p est le premier élément de la δ-chaîne d_{j+1}. De plus, tout élément situé entre d_j^d et d_{j+1}^p n'appartient pas à Σ.*

En considérant toujours le même exemple sur D, pour $\delta = 2$ on obtient la δ-chaîne maximale : $\{\{c, a, e, d\}, \{b, g, f\}\}$.

Définition 2.4 *Un sous-ensemble S de Σ est un δ-ensemble des chromosomes C et D si S est une δ-chaîne à la fois dans C et dans D.*

Prenons pour exemple les gènes de la figure 2.17. En considérant $\delta = 2$, on a $\{a, c\}$ qui est une δ-chaîne à la fois sur C et sur D ; donc $\{a, c\}$ est un δ-ensemble de C et D.

Définition 2.5 *Une δ-équipe sur les chromosomes C et D est un δ-ensemble maximal, c'est-à-dire un ensemble de δ-ensembles couvrant le plus de gènes possible sur C et D et dont l'intersection est nulle.*

Notation 2.6 *Une ligue sur les chromosomes C et D est l'union des équipes des chromosomes C et D.*

En utilisant ce formalisme, nous allons maintenant pouvoir étudier les deux algorithmes développés par Bergeron *et col.* (2003) pour découvrir les *équipes de gènes*[27]. Le premier d'entre eux est une approche polynomiale du problème.

[27] Genes Teams

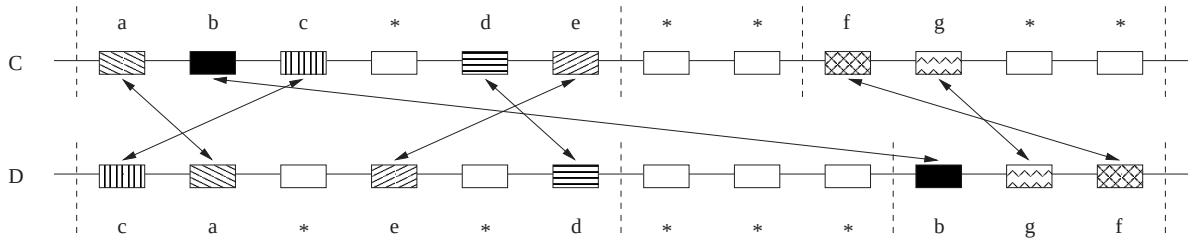


FIG. 2.17 – Conservation entre deux séquences de chromosomes C et D. Le "nom" de chaque gène est porté au-dessus de ce dernier dans le chromosome C (respectivement au-dessous dans le chromosome D). Les délimitations en pointillés indiquent les δ-chaînes maximales pour $\delta = 1$. Ainsi, sur C a-t-on $\{\{a, b, c, d, e\}, \{f, g\}\}$ et sur D : $\{\{c, a, e, d\}, \{b, g, f\}\}$.

Soient deux permutations sur Σ, π_C et π_D déjà partitionnées en δ-chaînes maximales sur les chromosomes C et D :

$$\pi_C = (c_1...c_{k_1})(c_{k_1+1}...c_{k_2})...(c_{k_s+1}...c_n)$$
$$\pi_D = (d_1...d_{l_1})(d_{l_1+1}...d_{l_2})...(d_{l_t+1}...d_n)$$

Soit $(c_i...c_j)$ une des classes de la partition de π_C. $(c_i...c_j)$ est une ligue. Le but de cet algorithme est de découper cette classe en m sous-classes $S_1, ..., S_m$ telles que :

- chaque sous-classe est une ligue,
- chaque sous-classe est une δ-chaîne dans C,
- chaque sous-classe est contenue dans une des classes de π_D.

Au début, il faut créer une sous-classe $S_1 = (c_i)$, puis il faut lire successivement les gènes c_k pour $i + 1 \leq k \leq j$. Lorsque l'on traite le gène c_k, considérant que les sous-classes S_1 à S_u ont déjà été créées, qu'elles sont toutes des δ-chaînes, et que chacune d'elles est contenue dans une des classes de π_D, alors :

- le gène c_k peut être ajouté comme le dernier élément d'une sous-classe déjà créée et dont le dernier élément est c, si et seulement si c et c_k appartiennent à une même classe dans π_D et que $\Delta_C(c, c_k) \leq \delta$. Sinon,
- le gène c_k commence une nouvelle sous-classe $S_u = (c_k)$.

L'algorithme répète ce processus alternativement sur les classes de π_C et π_D jusqu'à ce que des classes soient égales dans les deux permutations. Sa complexité est en $O(n^2)$, où n est le nombre de gènes.

Reprenons l'exemple de la figure 2.17 et déroulons quelques itérations de l'algorithme. Pour $\delta = 1$ nous avons :

$$\pi_C = (a\ b\ c * d\ e) * * (f\ g) * *$$
$$\pi_D = (c\ a * e * d) * * * (b\ g\ f)$$

- $S_1 = (a)$, on lit b :
  a et b appartiennent à une même classe dans π_C,
  a et b n'appartiennent pas à une même classe dans π_D, donc $S_2 = (b)$.

- $S_1 = (a)$, $S_2 = (b)$, on lit c :
  a et c appartiennent à une même classe dans π_C et dans π_D,
  $\Delta_C(a, c) \leq \delta$ dans π_C et dans π_D, donc $S_1 = (a, c)$.
- $S_1 = (a, c)$, $S_2 = (b)$, on lit d :
  c et d appartiennent à une même classe dans π_C et dans π_D,
  $\Delta_C(c, d) > \delta$ dans π_D, donc $d \notin S_1$;
  b et d n'appartiennent pas à une même classe dans π_D, donc $S_3 = (d)$.
- $S_1 = (a, c)$, $S_2 = (b)$, $S_3 = (d)$, on lit e :
  c et e appartiennent à une même classe dans π_C et dans π_D,
  $\Delta_C(c, e) > \delta$ dans π_C et dans π_D, donc $e \notin S_1$;
  b et e n'appartiennent pas à une même classe dans π_D, donc $e \notin S_2$.
  d et e appartiennent à une même classe dans π_C et dans π_D,
  $\Delta_C(d, e) \leq \delta$ dans π_C et dans π_D, donc $S_3 = (d, e)$.
- $S_1 = (a, c)$, $S_2 = (b)$, $S_3 = (d, e)$, etc.

Sur cet exemple, on obtient ainsi les équipes de gènes conservés :$\{\{a, c\}, \{b\}, \{d, e\}, \{f, g\}\}$.

En appliquant une stratégie du type "Diviser pour régner" sur cette méthode, Bergeron *et col.* (2003) ont développé un algorithme beaucoup plus rapide. Son principe est d'extraire de petites ligues à partir des ligues plus grosses. Soit S une ligue sur les chromosomes C et D. Les gènes de S sont respectivement ordonnés dans C et D comme $c_1...c_n$ et $d_1...d_n$. Si S est un δ-ensemble alors S est une δ-équipe. Si S n'est pas un δ-ensemble, il y a au moins deux éléments consécutifs c_i et c_{i+1} qui sont distants de plus de δ. Ainsi, $(c_1...c_i)$ et $(c_{i+1}...c_n)$ sont des ligues, coupant le problème initial en deux sous-problèmes. En partant du problème de la figure 2.17, on obtiendrait d'abord un partionnement du problème en deux sous-problèmes (1) et (2) :

$$\textbf{(1)} \quad \begin{array}{l}(a\ b\ c * d\ e) * * * * * * \\ (c\ a * e * d) * * * (b) * *\end{array} \qquad \textbf{(2)} \quad \begin{array}{l}* * * * * * * *\ (f\ g) * * \\ * * * * * * * * * * (g\ f)\end{array}$$

Le sous-problème (1) peut à son tour être décomposé en deux sous-problèmes (3) et (4) :

$$\textbf{(3)} \quad \begin{array}{l}(a * c * d\ e) * * * * * * \\ (c\ a * e * d) * * * * * *\end{array} \qquad \textbf{(4)} \quad \begin{array}{l}*(b) * * * * * * * * * * \\ * * * * * * * * * * *(b)\end{array}$$

La complexité devient $O(n \log^2(n))$ où n est le nombre de gènes. Cette méthode a été également adaptée à la comparaison de m génomes différents, sa complexité devenant $O(m\ n \log^2(n))$. Le cas des chromosomes circulaires, courant chez les bactéries, a été traité. Cette méthode a été appliquée pour l'étude de l'opéron tryptophane chez trois archebactéries (Luc *et col.*, 2003). Je reviendrai sur ces résultats dans la section suivante, lorsque j'effectuerai une comparaison des résultats obtenus par String, Gene Teams, et mon algorithme.

2.3 Résultats

Je m'attacherai ici à étudier le comportement des trois algorithmes détaillés précédemment.

2.3.1 STP

Pour permettre une utilisation simple ainsi qu'une analyse approfondie des résultats produits par l'algorithme basé sur les STP, j'ai développé une interface web. Par la suite, j'ai étudié l'impact des différents paramètres de cet algorithme. Pour finir je présenterai des résultats de problèmes particuliers validant notre approche.

2.3.1.1 Interface

Dans cette application, la relation génique employée est l'orthologie au sens de COG (Tatusov *et col.*, 1997). En effet, dans un premier temps nous avons calculé et utilisé des relations d'isorthologie mais les résultats n'étaient pas satisfaisants. L'isorthologie étant une relation très forte, seul un petit nombre de gènes était conservé, et dans le cas des transporteurs ABC, il ne s'agissait bien souvent que des transporteurs ABC eux-mêmes. De plus, le calcul des relations d'isorthologie par Blasts successifs était très long. Nous avons donc préféré utiliser les données de COG en l'enrichissant des nouveaux génomes à la manière de String[28].
L'interface a été travaillée de manière à être aussi simple et efficace que possible. Il faut tout d'abord sélectionner les génomes sur lesquels on désire travailler. En effet, suivant les cas, l'étude du voisinage d'un gène chez trois souches différentes d'une même bactérie ne sera pas informatif et ne contribuera qu'à densifier le volume de résultats. La seconde étape consiste à donner le nom du gène – ou le groupe COG – dont on souhaite explorer le voisinage. Il faut ensuite fixer la taille de la fenêtre puis la valeur maximale de l'écart de seuil entre gènes. Les données relatives aux gènes telles que l'orientation, la fonction, ou encore le groupe COG sont recherchées dans des fichiers plats issus d'une base de données de type AceDB (`http://www.acedb/org`). Après un laps de temps de quelques secondes[29], le résultat de l'exploration du voisinage du gène soumis (ici un gène du groupe COG 3839) apparaît à l'écran (Figure 2.18). Les gènes sont représentés par des flèches indiquant l'orientation relative du gène. Deux gènes d'une même couleur sont des gènes orthologues au sens de COG (et peuvent donc être paralogues) et conservés dans au moins deux voisinages. Les gènes grisés sont les gènes appartenant au même groupe COG que le gène d'ancrage et ceux marqués d'un point en leur centre ne possèdent pas de groupe COG. On retrouve ici tous les partenaires du transporteurs ABC et deux groupes COG, correspondant à des enzymes, sont conservés à proximité. A partir de ces résultats, on peut prédire que ces systèmes transportent du sucre. En cliquant sur un gène on peut accéder directement à toutes ses informations sur la base ABCdb.
Deux fenêtres complémentaires indiquent la fonction des groupes COG conservés et leur nombre d'occurrences, c'est-à-dire le nombre de fois où ils apparaissent dans le voisinage du groupe COG correspondant au gène d'ancrage (Figure 2.19), ainsi qu'un arbre de classification des gènes conservés en fonction de leur nombre d'occurrences (Figure 2.20). Cet arbre a été construit à l'aide du logiciel Phylip (Felsenstein, 1989) disponible sur le site :
`http://evolution.genetics.washington.edu/phylip.html`. Les programmes employés sont

[28] Les mises à jour de la base de données COG sont rares et il existe donc un grand nombre de génomes nouvellement séquencés qui sont absents.

[29] Les comparaisons s'effectuant deux à deux et prenant tour à tour chaque génome comme génome de référence, pour n génomes nous effectuons $n \times (n-1)$ appels à l'algorithme.

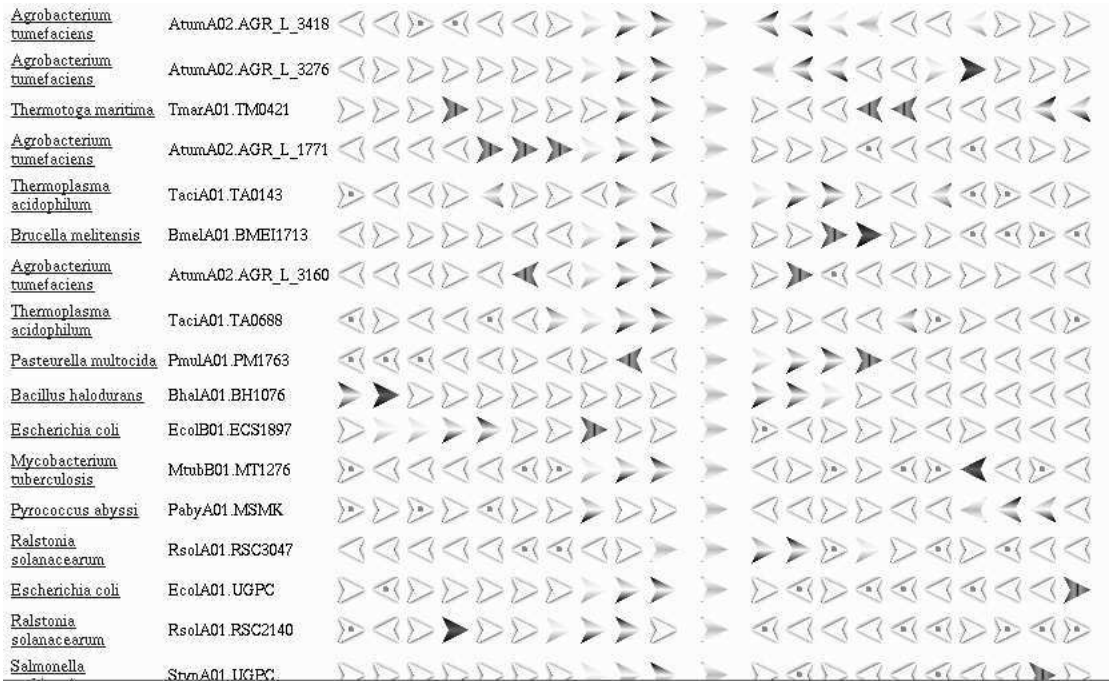


FIG. 2.18 – Exploration du voisinage des gènes du groupe COG3839 (ATPases d'un système de transport type-ABC). Les gènes sont représentés par des flèches indiquant l'orientation du gène, les motifs grisés indiquent une relation d'orthologie au sens de COG (qui peut donc être également une relation de paralogie), et le point dans un gène signale l'absence de classification dans un groupe COG. Les gène centraux gris clair sont les gènes orthologues au gène d'ancrage.

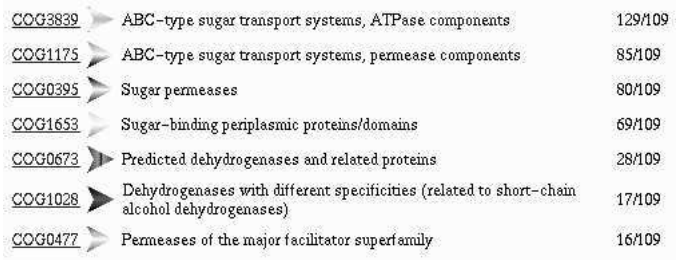


FIG. 2.19 – Fonction et nombre d'occurrences des groupes COG conservés. Pour chaque groupe COG affiché, un lien est disponible vers la base de données COG du NCBI.

Neighbor – utilisant la "*Neighbor Joining Method*" (Saitou et Nei, 1987) de distances matricielles – et *Drawgram* pour représenter l'arbre.
Les résultats présentés lors de cette étape sont très denses ; on rencontre par exemple des gènes dont le nombre d'occurrences est minimal (de 2), donc aucunement informatif. Il est alors possible de demander un affichage des COG conservés dont le nombre d'occurrences est supérieur à un seuil So déterminé par l'utilisateur ou bien automatiquement. Pour calculer ce seuil automatiquement, on classe les COG par nombre d'occurrences décroissant, ce qui donne une courbe en escalier du type de celle représentée en figure 2.21. On définit des "paliers" $P_1, ..., P_n$ qui représentent des ensembles de groupes COG ayant le même nombre d'occurences. Le palier P_i commence au groupe COG C_i et s'achève au COG C_{i+1} ; il a ainsi une longueur de $l_i = C_{i+1} - C_i$. Pour passer du palier P_i au palier P_{i+1} il faut effectuer un "saut" de longueur $s_i = o(P_i) - o(P_{i+1})$ où $o(x)$ est l'ordonnée du palier x. Le seuil correspond au palier pour lequel le saut est maximal et dont la longueur est inférieure à la longueur du palier suivant :

$$So = \{\exists k, P_k/s_k \text{ est maximal et } l_k < l_{k+1}\}$$

Dans le cas où, pour le saut maximal, la longueur est supérieure à la longueur du palier suivant, le seuil So correspondra au palier précédent de manière à ne perdre aucune information. So représente en général une bonne approximation du nombre minimal d'occurrences d'un gène conservé pour être informatif : les conservations les plus faibles sont supprimées. On obtient ainsi la liste des gènes conservés les plus significatifs. Ces résultats dépendent énormément des paramètres fixés : nous avons cherché à savoir quelles valeurs attribuer à ces paramètres pour obtenir des résultats optimaux (nombre de gènes conservés dépassant le nombre de gènes des transporteurs ABC déjà identifiés, et éviter de retrouver plusieurs transporteurs ABC au sein d'une même fenêtre).

2.3.1.2 Etude des paramètres de l'algorithme

Dans un premier temps, j'ai cherché à montrer l'intérêt de considérer tous les gènes du voisinage sans contrainte sur l'orientation. Puis, j'ai voulu déterminer la valeur δ de l'écart de seuil entre gènes qui était la plus significative. Pour cela, j'ai étudié le voisinage des ATPases des transporteurs ABC de 57 génomes (voir table 2.2) en fixant arbitrairement la taille de la fenêtre

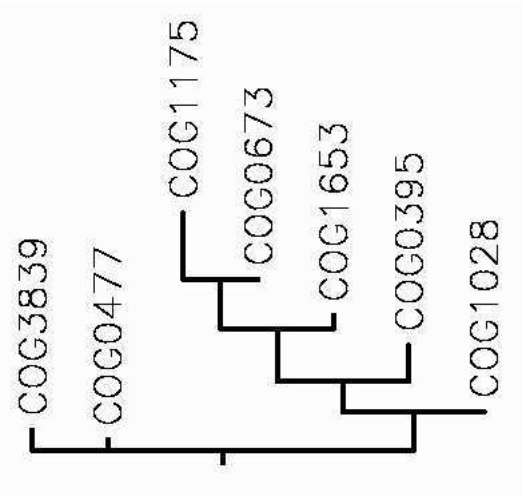


FIG. 2.20 – Arbre de classification des gènes conservés basé sur une table de contingence. Cet arbre permet d'évaluer les liens de proximités entre les gènes et d'en déduire des relations fonctionnelles. On représente des co-occurrences de gènes. Ici on est en présence d'une MSD (COG1175) très fortement liée à une enzyme (COG0673) : Ces deux gènes se retrouvent fréquemment associés dans un même voisinage.

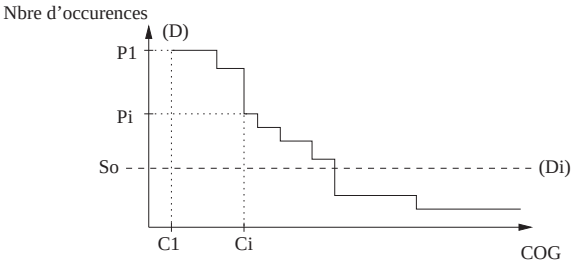


FIG. 2.21 – Calcul du seuil S_o – nombre minimal d'occurrences d'un gène conservé pour être très significatif.

Aeropyrum pernix	*Agrobacterium tumefaciens*
Aquifex aeolicus	*Archaeoglobus fulgidus*
Bacillus anthracis	*Bacillus halodurans*
Bacillus subtilis	*Borrelia burgdorferi*
Bradyrhizobium japonicum	*Brucella melitensis*
Buchnera sp.	*Campylobacter jejuni*
Caulobacter crescentus	*Chlamydia muridarum*
Chlamydia trachomatis	*Chlamydophila pneumoniae*
Clostridium acetobutylicum	*Deinococcus radiodurans*
Enterococcus faecalis	*Escherichia coli*
Haemophilus influenzae	*Halobacterium sp.*
Helicobacter hepaticus	*Helicobacter pylori*
Lactococcus lactis	*Listeria innocua*
Mesorhizobium loti	*Methanocaldococcus jannaschii*
Methanosarcina acetivorans	*Methanosarcina mazei*
Methanothermobacter thermautotrophicus	*Mycobacterium leprae*
Mycobacterium tuberculosis	*Mycoplasma genitalium*
Mycoplasma pneumoniae	*Neisseria meningitidis*
Nostoc sp.	*Pseudomonas aeruginosa*
Pyrococcus abyssi	*Pyrococcus furiosus*
Pyrococcus horikoshii	*Rhizobium sp.*
Rickettsia conorii	*Rickettsia prowazekii*
Salmonella enterica	*Sinorhizobium meliloti*
Streptomyces coelicolor	*Sulfolobus solfataricus*
Sulfolobus tokodaii	*Synechocystis sp.*
Thermoplasma acidophilum	*Thermoplasma volcanium*
Thermotoga maritima	*Treponema pallidum*
Ureaplasma urealyticum	*Vibrio vulnificus*
Xylella fastidiosa	

Tab. 2.2 – Liste des 57 génomes étudiés.

à 10. Les génomes considérés ont été sélectionnés de manière à n'avoir qu'une seule souche par espèce et n'avoir que des espèces taxonomiquement éloignées. Un biais important est ainsi supprimé. En effet, lors de l'étude du voisinage d'un gène d'une souche d'*Escherichia coli* par exemple, on retrouvera ce voisinage à peu près intégralement dans les trois autres souches de la même espèce ou bien encore chez *Salmonella enterica* qui est très proche d'*Escherichia coli* taxonomiquement. On supprime ainsi une redondance d'informations inutiles. De plus, dans des espèces taxonomiquement éloignées, les génomes ont été complètement remaniés : une conservation de gènes indique donc une conservation de fonction d'autant plus forte. Les résultats sont présentés sur la figure 2.22. Il s'agit du nombre moyen de gènes conservés en fonction de l'écart de seuil entre gènes. Les augmentations pour les grandes valeurs de δ ne sont probablement dues qu'aux paires de génomes taxonomiquement proches. Ces résultats montrent bien que le fait de ne considérer aucune contrainte sur l'orientation permet de détecter en moyenne de 1 à 2 gènes de plus (ce qui représente 20% de la conservation totale pour $\delta = 1$). Le nombre moyen de gènes conservés augmente très fortement jusqu'à $\delta = 2$ puis plus faiblement jusqu'à $\delta = 10$. Il se stabilise beaucoup plus rapidement lorsque l'on ne considère que les gènes de même orientation. La variance, elle, diminue progressivement, quelle que soit la courbe : en relachant les contraintes, on conserve beaucoup plus d'éléments dans tous les cas de figure et les écarts sont donc plus faibles. Le meilleur compromis entre le nombre moyen de gènes conservés (sans tenir compte des espèces trop proches taxonomiquement) et la confiance des prédictions fonctionnelles semble être $\delta = 3$. C'est donc cette valeur que nous utiliserons par la suite.

Nous avons fixé la taille de la fenêtre à $w = 10$ car biologiquement il n'y a pas de conservation pour un éloignement trop important.

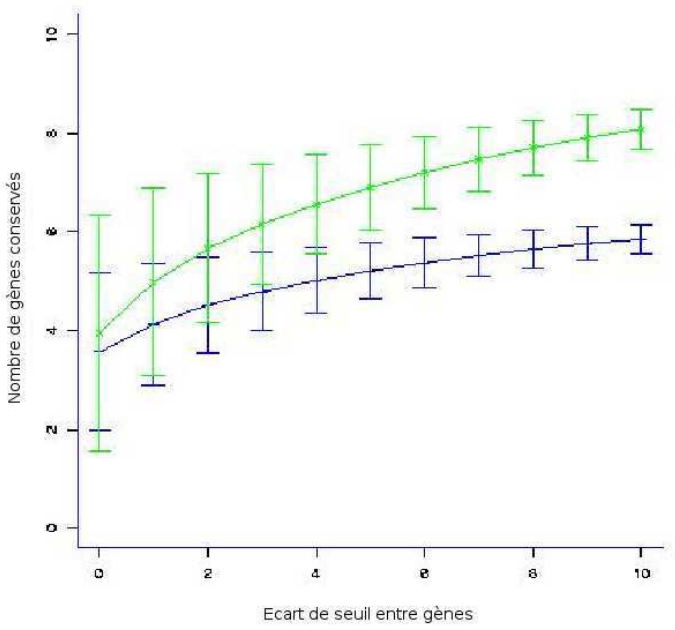


FIG. 2.22 – Courbes de conservation moyenne du nombre de gènes en fonction de l'écart de seuil entre gènes (en bleu on ne considère que les gènes orientés dans le même sens, et en vert, l'orientation est quelconque). La variance est indiquée en chaque point. Les gènes d'ancrage sont les domaines NBD des 57 génomes étudiés (importeurs et exporteurs). On peut remarquer qu'un transporteur ABC est en général composé de 2 ou 3 gènes, donc seules les conservations supérieures sont pertinentes. Le nombre de gènes conservés ne peut pas diminuer, contrairement au nombre de paires de génomes qui diminue rapidement : on a de moins en moins d'information. Après $\delta = 3$ ce ne sont plus les liens fonctionnels qui doivent être détectés mais plutôt des liens d'ordre taxonomique.

Aeropyrum pernix	*Aquifex aeolicus*
Bacillus subtilis	*Borrelia burgdorferi*
Campylobacter jejuni	*Deinococcus radiodurans*
Escherichia coli	*Helicobacter pylori*
Lactococcus lactis	*Mesorhizobium loti*
Methanothermobacter thermautotrophicus	*Mycoplasma genitalium*
Neisseria meningitidis	*Pseudomonas aeruginosa*
Pyrococcus abyssi	*Sulfolobus solfataricus*
Thermoplasma volcanium	*Thermotoga maritima*
Treponema pallidum	*Xylella fastidiosa*

Tab. 2.3 – Liste des 20 génomes étudiés.

2.3.1.3 Relation entre une famille de régulateurs et une famille de transporteurs ABC

En étudiant les transporteurs ABC de la sous-famille A_5, nous avons remarqué qu'ils étaient souvent associés à la famille de régulateurs *LacI*. Nous avons donc adopté une démarche inverse à la précédente, nous ne nous sommes plus ancrés sur un gène codant pour un domaine d'un transporteur ABC mais sur un régulateur transcriptionnel et nous avons cherché à savoir s'il était toujours associé à la même famille de transporteurs ABC. Le régulateur choisi est le gène *lacI* d'*Escherichia coli* (groupe COG1609) qui appartient à une famille multigénique. Le nombre de génomes a été réduit à 20 taxonomiquement éloignés (voir table 2.3) pour présenter des résultats plus facilement interprétables. Nous avons obtenu 71 voisinages conservés correspondant à 1491 gènes parmi lesquels 241 – soit 16% – ne possédaient pas de groupe COG connu. Dans 50% des voisinages il y avait au moins un gène de transporteur ABC. Des résultats de voisinage obtenus, nous avons tiré un tableau montrant les co-occurrences de gènes codant pour des transporteurs ABC (voir table 2.4). Sur la première ligne on peut ainsi lire que le COG 1172, correspondant à une MSD de la famille A_1 des transporteurs ABC, apparaît dans 31 voisinages et qu'il est conservé 30 fois en co-occurrence avec les groupes 1129 et 1879, et 4 fois avec 0687. D'après ces groupes COG associés à des sous-familles on peut déduire une sous-classification de ces sous-familles.

2.3.2 Différences entre les trois méthodes

Les paramètres de mon algorithme étant fixés ($w = 10$, $\delta = 3$), nous allons pouvoir comparer les résultats obtenus grâce aux trois méthodes. L'étude de GeneTeams (Luc *et col.*, 2003) ne portant que sur l'opéron tryptophane chez trois archaebactéries (*Archeoglobus fulgidus*, *Methanococcus thermoautotrophicum* et *Pyrococcus abyssi*), je prendrai comme gène d'ancrage pour ma méthode et pour String le gène trpA d'*Archeoglobus fulgidus*[30]. Pour illustrer les différences de comportement, j'ai ajouté un génome supplémentaire où les gènes de l'opéron tryptophane ont été remaniés[31] : *Aeropyrum pernix*. Pour en simplifier la lecture, les résultats obtenus dans le tableau 2.5 ne comportent pas les gènes sans relation qui sont intercalés. Le seuil choisi pour

[30] Il s'agit du premier gène de l'opéron. Les résultats sont identiques en considérant comme gène d'ancrage un autre gène de l'opéron.

[31] Nous sommes en présence de gènes dans les deux sens transcriptionnel.

	Famille	1172	1129	1879	0747	1123	0601	1173	1653	1175	3839	3834	3833	0687	1131
COG1172	M_1	**31**	**30**	**30**	–	–	–	–	–	–	–	–	–	4	–
COG1129	N_1	**30**	**31**	**32**	–	–	–	–	2	2	2	–	–	4	–
COG1879	S_1	**30**	**32**	**31**	–	–	–	–	2	2	2	–	–	4	–
COG0747	S_2	–	–	–	**7**	**2**	**4**	**4**	4	4	4	–	–	–	–
COG1123	N_2	–	–	–	**2**	**1**	**2**	**2**	2	2	2	–	–	–	–
COG0601	M_2	–	–	–	**4**	**2**	**3**	**4**	2	2	2	–	–	–	–
COG1173	M_2	–	–	–	**4**	**2**	**4**	**3**	2	2	2	–	–	–	–
COG1653	S_5	–	2	2	4	2	2	2	**23**	**24**	**10**	–	–	–	2
COG1175	M_5	–	2	2	4	2	2	2	**24**	**23**	**10**	–	–	–	2
COG3839	N_5	–	2	2	4	2	2	2	**10**	**10**	**9**	–	–	–	–
COG3834	M_5	–	–	–	–	–	–	–	–	–	–	**7**	**8**	–	2
COG3833	M_5	–	–	–	–	–	–	–	–	–	–	**8**	**7**	–	2
COG0687	S_5	4	4	4	–	–	–	–	–	–	–	–	–	**5**	–
COG1131	N_7	–	–	–	–	–	–	–	2	2	–	2	2	–	**3**

TAB. 2.4 – Etude des transporteurs ABC conservés au voisinage du régulateur transcriptionnel *lacI* (COG1609). Chaque ligne indique les co-occurrences de conservation de groupes COG ainsi que le nombre d'occurrences. Pour chaque famille de transporteurs A_1, A_2, A_5 et A_7, sont indiquées en gras les COG compatibles (les partenaires permettant de constituer un transporteur fonctionnel).

GeneTeams							
A. per.	-trpG	-trpE	-trpD	trpA			
A. ful.	trpD	trpE	trpG	trpF	trpB	trpA	
M. ther.	trpE	trpG	trpC	trpF	trpB	trpA	trpD
P. aby.	trpC	trpD	trpE	trpG	trpF	trpB	trpA
String							
A. per.	trpA	trpB					
A. ful.	trpC-D	trpE	trpG	trpF	trpB	trpA	
M. ther.	trpE	trpG	trpC	trpF	trpB	trpA	trpD
P. aby.	trpC	trpD	trpE	trpG	trpF	trpB	trpA
STP							
A. per.	-trpG	- trpE	-trpD	trpA	trpB	trpC	
A. ful.	trpD	trpE	trpG	trpF	trpB	trpA	
M. ther.	trpE	trpG	trpC	trpF	trpB	trpA	trpD
P. aby.	trpC	trpD	trpE	trpG	trpF	trpB	trpA

Tab. 2.5 – Comparaison des résultats obtenus par les trois méthodes en prenant le gène *trpA* comme gène d'ancrage.

ma méthode et GeneTeams est $\delta = 3$; une différence d'orientation des gènes est indiquée par un signe '-'.
Les résultats ne diffèrent que sur le premier génome. Tout d'abord String ne détecte que deux gènes conservés. En effet les trois gènes trpG, trpE, et trpD sont en orientation inverse. Les gènes trpC et trpD ne sont pas reconnus, contrairement à la méthode STP : le gène trpD est orienté en sens inverse et n'est donc pas détecté ; le gène trpC est trop éloigné (il rompt la série au sens d'Overbeek *et col.* (1999)).
D'autre part, String tient compte de la fusion des gènes (notamment pour le gène trpC-D). C'est la raison pour laquelle sur *Archeoglobus fulgidus* String est le seul à pouvoir détecter le domaine trpC.
Pour les différences de conservation entre les méthodes GeneTeams et STP, la conservation de trpB et trpC dans le premier génome s'explique très simplement : ces gènes sont absents au moins une fois dans l'un des génomes considérés. Comme GeneTeams recherche des groupes de gènes conservés dans tous les génomes, il considèrera forcément que ces gènes sont conservés sous la forme de gènes isolés et ne détectera donc pas la conservation générale.
Dans ce cas précis, on a une perte d'information sur *Aeropyrum pernix* en utilisant String ou GeneTeams ; la méthode STP semble donc être plus adaptée à l'étude des transporteurs ABC.

2.4 Conclusion & Perspectives

Pour rechercher un voisinage conservé, on doit tout d'abord être capable d'identifier les gènes

conservés entre génomes. Ceci est effectué en détectant les relations d'orthologie. Nous accordons beaucoup de poids à cette notion puisqu'on admet généralement que des gènes orthologues codent pour la même fonction dans des génomes différents. Mais peut-être peut-on se passer de cette relation lorsque l'on recherche une conservation de gènes. En effet, la co-occurrence de gènes homologues peut être suffisamment informative pour suspecter des liens fonctionnels. Une illustration de cette idée est donnée par l'identification de l'association fréquente de gènes codant pour des systèmes à deux composants avec des gènes codant pour des transporteurs ABC dans le groupe *Bacillus/Clostridium* ; des liens fonctionnels ont été découverts au moins au niveau transcriptionnel (Joseph *et col.*, 2002). Inversement, une relation plus "dure" telle que l'isorthologie ne donne pas de bons résultats car beaucoup trop discriminante. Il faudrait donc pouvoir sélectionner une relation d'homologie parmi une liste lors de la requête. L'utilisation des données de la base HobacGen (Perrière *et col.*, 2000) pourrait être une piste intéressante. En effet, dans le cas des familles multigéniques (tel que *lacI*), il existe des sous-groupes apparaissant sur les arbres phylogéniques. L'utilisation de ces informations pourrait permettre de valider les prédictions de conservation entre régulateur et transporteur.
Il y a également le cas des gènes ayant fusionné. Pour l'instant, ils ne sont pas pris en compte par l'algorithme STP (bien que présents dans la base COG) et sont considérés comme un unique gène possédant donc un orthologue unique. Une modification très simple de l'algorithme permettrait d'intégrer de tels gènes et d'obtenir ainsi des résultats plus précis[32]. Cette modification entraînant également une reprogrammation de l'interface, elle n'a pas encore été effectuée.
On peut aussi utiliser des méthodes compémentaires qui vont apporter des informations différentes. Par exemple, une analyse par régressions linéaires simples a été réalisée pour la famille A_5 et son voisinage[33] (Nicolas, 2003). Cette analyse, basée sur les grandes fonctions biologiques des groupes COG, a montré que la fonction des gènes représentés dans le voisinage des importeurs de la famille A_5 serait liée au métabolisme dans lequel est impliqué le substrat importé.
Cet algorithme rapide peut traiter de nombreux génomes et son interface permet une visualisation précise des résultats. Toutefois, une amélioration possible serait la détection automatique de liens fonctionnels. Cette étape ferait appel à un module de fouille de textes pour déterminer si la fonction COG du gène conservé est compatible ou non. Il devrait également être intégré dans la base de connaissance ABCkb (Capponi *et col.*, 2001) et ainsi l'enrichir automatiquement avec les informations prédites.

[32] Cette modification a d'ailleurs été apportée très récemment à l'algorithme GeneTeams (Pasek *et col.*, 2004)
[33] Fréquence des groupes COG dans le voisinage de la famille A_5 en fonction de leur fréquence dans les génomes.

3

Construction de classes de systèmes intégrés

Il n'y a pas de grande tâche difficile qui ne puisse
être décomposée en petites tâches faciles.
Dilgo Khyentse Rinpoché

Les transporteurs ABC, chez les procaryotes, sont impliqués dans les échanges entre la bactérie et le milieu extérieur. Ils transportent une grande variété de substrats et peuvent être classés en sous-familles sur la base de similarités de séquences. Ces familles sont associées à de grands types de substrats (ions, sucres, acides aminés) comme le montre le tableau 3.6.

Importeurs		**Exporteurs**	
Famille	**Substrat**	**Famille**	**Substrat**
A_1	Sucres de type ribose	A_3	Résistance aux macrolides
A_2	Oligopeptides	A_6	Multidrogues résistance
A_4	Acides aminés	A_7	Résistance/Export d'antibiotique
A_5	Oligosaccharide/Glycine/Bétaïne	A_9	Résistance/Export d'antibiotique
A_8	Sidérophores (Fer, Zinc)	A_{12}	Substrat inconnu
A_{10}	Acides aminés branchés	A_{13}	Substrat inconnu
A_{11}	Phosphate	A_{14}	YurY ?
A_{17}	Molybdate ?	A_{15}	Seulement dans les archeae ?
A_{18}	Phosphonate	A_{16}	Export d'hème
		A_{19}	Substrat inconnu

Tab. 3.6 – Classification des transporteurs ABC d'après Quentin *et col.* (2002).

Ces familles ne permettent pas de prédire avec précision le substrat d'un transporteur ABC, et sont même, dans certains cas, de substrat totalement inconnu (comme les familles A_{12}, A_{13}, A_{15}

Sous-familles de A_5	**Substrat transporté**
N_{5a}	Sucres
N_{5b}	Glycine/Bétaïne
N_{5c}	Nitrates
N_{5d}	Spermidine/Putrescine
N_{5f}	Sulfate

Tab. 3.7 – Classification en sous-familles de la classe A_5 d'après Quentin *et col.* (2002).

et A_{19}). Toutefois, certaines familles, telle que A_5, sont découpées en sous-familles sur la base de similarité de séquences des MSD (tableau 3.7). Il faut noter que le substrat indiqué pour chaque sous-famille n'est pas clairement établie : des éléments de N_{5d} sont par exemple impliqués dans le transport de sulfate.
Si nous admettons que les systèmes transportant le même substrat sont codés par des gènes orthologues, alors le fait de constituer des groupes de gènes orthologues (Tatusov *et col.*, 2001) permet d'obtenir une indication fonctionnelle précise. Pour obtenir une prédiction plus fiable, il est préférable de restreindre la relation d'orthologie à l'isorthologie[34] (Fitch, 2000). De cette relation, nous construisons un graphe Γ dont les sommets représentent les protéines des génomes considérés. Comme l'orthologie est une relation transitive, les gènes isorthologues deux à deux devraient constituer des sous-graphes complets, c'est-à-dire des cliques de Γ. En pratique, en raison principalement de bruits et d'erreurs sur l'estimation des distances évolutives, et du fait de l'utilisation de la notion de "meilleur score" qui ne désigne pas nécessairement un orthologue, les parties connexes ne sont généralement pas des sous-graphes complets disjoints et différents groupes d'isorthologues peuvent se retrouver dans la même classe par la présence de liens artéfactuels. Il est donc nécessaire de rechercher des zones denses dans Γ, c'est-à-dire des classes de sommets qui présentent un fort pourcentage d'arêtes internes. Ce sont ces *quasi-cliques* (Matsuda *et col.*, 1999) qui devraient constituer des classes d'isorthologie – et donc des familles de transporteurs ABC caractéristiques du substrat transporté. Je rappellerai dans ce chapitre les notions de base de la classification, puis j'exposerai la méthode de recherche de zones denses dans un graphe que nous avons développée. Je présenterai ensuite une méthode de classification concurrente qui cherche à structurer un graphe en "communautés" (Girvan et Newman, 2002). Enfin, je décrirai la validation de l'algorithme de recherche des classes denses sur des graphes aléatoires et les résultats obtenus sur les transporteurs ABC.

3.1 Généralités sur la classification

Le problème de la classification est de définir des sous-ensembles d'objets appelés *classes* ou *groupes*. Ces classes peuvent être disjointes ou chevauchantes, voire emboîtées suivant la finalité de la méthode. Les classes possibles ne sont pas connues à l'avance. Le but est alors de regrouper au sein d'un même groupe les objets considérés comme similaires afin de constituer les classes.

[34] Les domaines NBD des transporteurs ABC appartenant à une famille hautement paralogue, le fait de considérer l'orthologie pourrait induire des chaînes assemblant différentes classes.

Le problème nécessite de définir une mesure de proximité entre objets, qui peut être estimée à l'aide d'une fonction.

3.1.1 Mesure de proximité

On utilise une mesure qui est définie sur les paires d'objets i et j ; on peut distinguer les *indices de similarité* qui mesurent la ressemblance entre les objets i et j, notés s_{ij}, et les *indices de dissimilarité* qui mesurent la dissemblance entre les objets i et j, notés d_{ij}. Ces indices ont les propriétés suivantes (Chandon et Pinson, 1981) :

- Un indice de proximité associe à chaque paire d'objets d'un ensemble O un nombre non négatif (propriété de non-négativité) :

$$\forall i, j \in O, s_{ij} \geq 0 \text{ ou } d_{ij} \geq 0$$

- La proximité entre deux objets est symétrique :

$$\forall i, j \in O, s_{ij} = s_{ji} \text{ ou } d_{ij} = d_{ji}$$

- Une dissimilarité est dite propre si et seulement si :

$$\forall i, j \in O, d_{ij} = 0 \Leftrightarrow i = j$$

  Un indice de dissimilarité propre qui vérifie les propriétés de non-négativité et de symétrie est appelé *indice de distance*.
- Dans un espace métrique, le chemin allant directement d'un objet à un autre est plus court qu'en passant par un troisième objet. Cette propriété d'inégalité triangulaire s'applique à un indice de dissimilarité qui devient alors une *distance* encore appelée *métrique*.

$$\forall i, j, k \in O, d_{ij} \leq d_{ik} + d_{jk}$$

- Enfin, l'inégalité ultramétrique, portant encore une fois essentiellement sur les indices de dissimilarité, permet de définir une *distance ultramétrique*.

$$\forall i, j, k \in O, d_{ij} \leq max(d_{ik}, d_{jk})$$

  L'inégalité ultramétrique implique l'inégalité triangulaire. Elle signifie que pour tout triplet i, j, k, les deux plus grandes valeurs de distances sont égales.

La connaissance des proximités entre paires d'objets n'est pas très "lisible" : il faut interpréter cette information par une représentation plus synthétique faisant apparaître une structuration des objets ; c'est l'objectif de la classification. Il existe de très nombreuses méthodes ((Jain *et col.*, 1999), (Chandon et Pinson, 1981)) dont la finalité peut être la recherche de classes, d'une hiérarchie de classes, de partitions (classification de O en classes disjointes), ou encore la recherche d'un recouvrement (classification de O en classes éventuellement chevauchantes). Je m'intéresserai ici plus particulièrement aux méthodes hiérarchiques et aux méthodes de partitionnement qui permettront d'introduire la méthode présentée par la suite.

3.1.2 Méthodes hiérarchiques

L'objectif des méthodes hiérarchiques est la recherche d'une famille de classes qui forment une hiérarchie qui peut être indicée.

Définition 1.1 *On appelle hiérarchie sur O un ensemble de classes $\mathcal{H}$ vérifiant :*

$$\begin{array}{ll} (i) & O \in \mathcal{H} \\ (ii) & \forall i \in O, \{i\} \in \mathcal{H} \\ (iii) & \forall H, H' \in \mathcal{H}, H \cap H' \in \{H, H', \emptyset\} \end{array}$$

Le (iii) signifie que deux classes de la hiérarchie sont soit disjointes soit emboîtées et que toute classe est la réunion de classes d'un niveau inférieur. A une hiérarchie $\mathcal{H}$ correspond un arbre dont les sommets sont les classes de $\mathcal{H}$ et les arêtes marquent la relation d'inclusion.

Définition 1.2 *Soient $\mathcal{H}$ une hiérarchie sur O, et une fonction f, appelée indice de niveau, $f : \mathcal{H} \mapsto \mathbb{R}^+$. On appelle hiérarchie indicée $(\mathcal{H}, f)$:*

$$\begin{array}{ll} (i) & \forall \{i\}, f(\{i\}) = 0 \\ (ii) & \forall H, H' \in \mathcal{H}, H \subset H' \Rightarrow f(H) < f(H') \end{array}$$

L'arbre représentant la hiérarchie, indicé par la fonction f, est appelé son *dendrogramme*. Le sommet de plus haut niveau, correspondant à l'ensemble O, définit la *racine* de l'arbre. Cet arbre est dit binaire si chaque sommet de niveau supérieur à 0 est la réunion de deux classes. Dans le cas d'une distance ultramétrique U, pour tout i, j, on note C_{ij} la classe de plus petit cardinal contenant i et j. La fonction f définie par $f(C_{ij}) = U(i, j)$ est une représentation exacte de U.

Les méthodes hiérarchiques sont les méthodes de construction d'un dendrogramme à partir d'une distance d, qui sont donc des approximations de d par une distance ultramétrique. Les plus connues des méthodes hiérarchiques sont les méthodes ascendantes et descendantes.

Classification hiérarchique ascendante : La construction de la hiérarchie s'obtient en fusionnant à chaque étape les deux classes les plus proches. Suivant la fonction choisie pour mesurer la distance entre classes, on obtient différentes méthodes, dont les célèbres lien unique, lien moyen, et lien complet. Dans la méthode du lien unique on définit la distance entre deux classes C_i et C_j par la plus petite valeur de distance les séparant :

$$d(C_i, C_j) = min_{k \in C_i, k' \in C_j}(d_{kk'})$$

Dans le lien moyen cette fonction minimum est remplacée par la moyenne et dans le lien complet par le maximum.

Classification hiérarchique descendante : La construction de la hiérarchie s'obtient en choisissant à chaque étape une classe que l'on subdivise en deux sous-classes. Certaines méthodes nécessitent d'étudier toutes les subdivisions binaires possibles ; dans ce cas, la méthode n'est pas polynomiale. D'autres minimisent des critères (diamètre, ...) en utilisant des algorithmes polynomiaux (Guénoche *et col.*, 1991).

3.1.2.1 Une méthode récente : Girvan et Newman (2002)

Girvan et Newman ont développé une méthode de classification originale. Elle s'applique à la recherche de classes de sommets dans un graphe et n'utilise pas directement la notion de distance. C'est une méthode hiérarchique descendante qui a la même finalité que celle que nous présentons ultérieurement, c'est-à-dire la recherche de groupes de sommets qui sont fortement connectés et qui possèdent peu de connexions entre eux. Ces auteurs utilisent le terme de "communautés" au lieu de "classes" à cause de leur premier domaine d'application : l'étude de relations entre personnes (réseaux sociaux). Cette méthode est intéressante par son approche novatrice. Elle a été testée sur des graphes aléatoires, ce qui en fait une bonne méthode comparative.
L'idée de cet algorithme est de pondérer chaque arête par le nombre de plus courts chemins qui la traverse et de déconnecter progressivement le graphe en choisissant à chaque itération l'arête de plus forte valeur. Intuitivement si plusieurs arêtes lient des classes fortement connectées, ce sont celles-ci qui seront de plus fort poids et devront être éliminées.

Définition 1.3 *Notons $[z,t]$ l'ensemble des plus courts chemins entre z et t. On appelle indice de liaison*[35] *d'une arête (x,y) le nombre de plus courts chemins entre paires de sommets qui passent par celle-ci :*

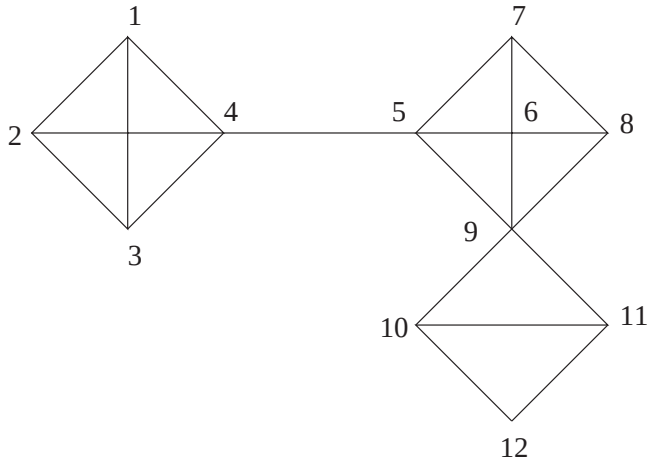
$$B(x,y) = \sum_{z,t} |ch \in [z,t] \text{ tel que } (x,y) \in ch|$$

S'il existe des plus courts chemins différents pour une même paire de sommets, ils seront tous considérés[36]*.*

Si un graphe contient des groupes qui ne sont liés que par quelques arêtes, alors les plus courts chemins entre des éléments des différents groupes doivent passer par ces arêtes. Elles auront donc une forte valeur de liaison. L'algorithme va couper ces arêtes pour dégager les classes : il s'agit d'une méthode divisive dans laquelle les arêtes seront progressivement retirées du graphe jusqu'à ce qu'il n'y en ait plus. Les étapes de cet algorithme sont :

1. On calcule l'indice de liaison de toutes les arêtes du graphe (pour m arêtes et n sommets, opération réalisée en $O(mn)$ en utilisant l'algorithme de (Newman, 2001)).
2. On retire l'arête de plus fort poids.
3. On recalcule l'indice de liaison pour toutes les arêtes qui appartenaient à la même composante connexe que l'arête retirée.
4. On répète depuis l'étape 2 jusqu'à ce qu'il ne reste plus d'arête.

[35] betweeness dans le texte, noté B
[36] Il s'agit du nombre de chemins et non pas du nombre de paires.



Arêtes	**1-2**	**1-3**	**1-4**	**2-3**	**2-4**	**3-4**	**4-5**	**5-6**	**5-7**	**5-9**	**6-7**	**6-8**
Indice de liaison	1	1	12	1	12	12	44	6	10	25	6	6

6-9	**7-8**	**8-9**	**9-10**	**9-11**	**10-11**	**10-12**	**11-12**
10	6	10	21	21	1	12	12

FIG. 3.23 – Graphe de 12 sommets et 20 arêtes ; la table indique l'indice de liaison de chaque arête.

Le résultat de cette classification peut être représenté comme un arbre hiérarchique. Si l'on désire se limiter à un nombre de classes, il faut déterminer un seuil.

Exemple : Nous pouvons calculer les indices de liaison du graphe de la figure 3.23. Nous voyons ici que l'arête de poids le plus fort correspond à $(4, 5)$ avec une valeur de 44. En effet, partant des 4 sommets 1 à 4 pour aller vers les 8 sommets 5 à 12, il y a $4 \times 8 = 32$ chemins qui passent par l'arête $(4, 5)$. Mais le chemin $[5, 8]$ peut être effectué de 3 manières différentes : $[5, 6] + [6, 8]$, ou $[5, 7] + [7, 8]$, ou encore $[5, 9] + [9, 8]$; le chemin $[9, 12]$ peut être décomposé en $[9, 10] + [10, 12]$ ou $[9, 11] + [11, 12]$. Ce qui ajoute 12 chemins, soit 44 chemins passant par l'arête $(4, 5)$. Cette arête sera supprimée lors de la première étape. Les indices de liaison sont alors recalculés pour déterminer la nouvelle arête à supprimer. En appliquant l'algorithme, on obtient comme résultat final l'arbre hiérarchique présenté en figure 3.24.

Cet algorithme, bien que donnant de bons résultats sur des configurations de graphes très spécifiques (nombre d'arêtes inter-classes très faible), présente des inconvénients :

- il n'y a aucune indication sur la valeur de seuil permettant de couper l'arbre et d'obtenir le "bon" nombre de classes.
- il est très lent : pour un graphe de n sommets et m arêtes, le calcul de l'indice de liaison des arêtes par l'algorithme de Newman (2001) se fait en $O(mn)$. Comme il faut effectuer cette opération à chaque fois que l'on retire une arête, on obtient $O(m^2 n)$.

Radicchi *et col.* (2003) ont proposé une amélioration pour détecter les classes plus rapidement. La

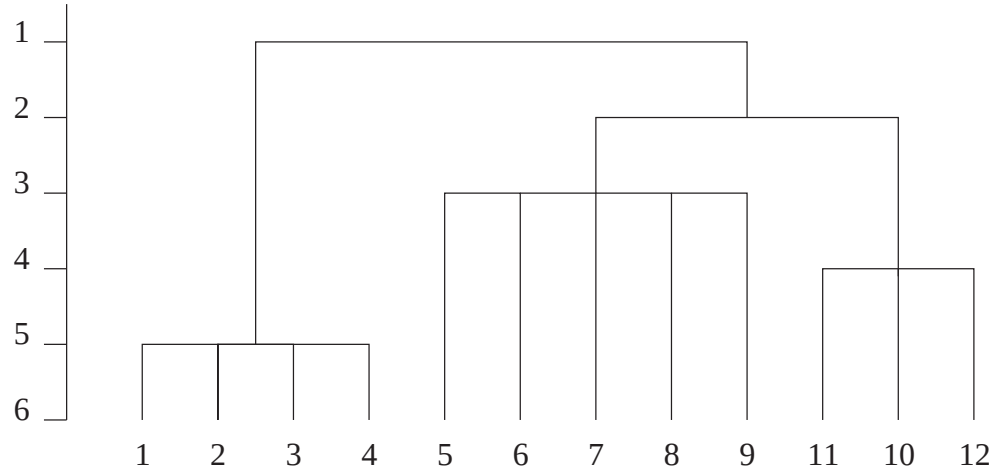


FIG. 3.24 – Arbre hiérarchique obtenu par l'algorithme des structures en communautés (Girvan et Newman, 2002) d'après le graphe de la figure 3.23. Le dendrogramme est indicé ici par l'ordre d'itération, ce qui ne fait nullement apparaître les symétries des sous-graphes.

mesure de l'indice de liaison est cette fois locale et peut être recalculée rapidement. La complexité de l'algorithme devient alors $O(m^4/n^2)$. Le nombre de classes reste toutefois à déterminer comme dans la méthode initiale.

3.1.3 Méthodes de partitionnement

Le but des méthodes de partitionnement est de construire une partition des éléments en q classes où le nombre q de classes est soit spécifié a priori, soit déterminé par la méthode.

Définition 1.4 *Soit S un ensemble à n éléments. Une partition de S est un ensemble de classes disjointes dont l'union est S. On note $P = \{S_1, S_2, ..., S_q\}$ une partition de S en q classes.*

$$\forall i, j \text{ on a } S_i \cap S_j = \emptyset \text{ et } \bigcup_{i=1,...,q} S_i = S$$

Il y a deux familles de méthodes de partitionnement couramment utilisées : les méthodes d'allocation/recentrage type "K-means" (MacQueen, 1967) ou "nuées dynamiques" (Diday, 1971), et les méthodes d'optimisation d'un critère sur l'ensemble des partitions à nombre de classes fixé.

Méthode d'allocation/recentrage : L'idée centrale des méthodes d'allocation/recentrage est de déterminer un ensemble de centres des classes et d'affecter les éléments au centre (et à la classe) dont ils sont le plus proche. Ces méthodes comprennent en général deux étapes :

– La génération de la configuration initiale : un ensemble de centres est choisi pour initialiser les q classes. Ils peuvent être aléatoirement choisis ou construits ; la partition résultante

dépend de la configuration initiale.
- Une boucle itérative d'allocation/recentrage : les itérations s'arrêtent lorsqu'un critère (généralement l'inertie : somme des carrés des écarts au centre) ne décroît plus.

Si les objets sont décrits dans un espace de représentation engendré par des variables, les centres des classes sont généralement choisis comme des centres de "gravité". Mais on peut aussi étendre ces méthodes à la simple donnée d'une distance, par exemple, en prenant pour centre une médiane ; on est alors dans la seconde famille.

Méthode d'optimisation d'un critère On cherche ici à construire une partition à nombre de classes fixé qui optimise un certain critère. Compte tenu de la nature discrète de l'ensemble des partitions, il s'agit d'optimisation combinatoire dans laquelle la matrice de distances est considérée comme un graphe complet pondéré par les valeurs de distance ; pour un survol récent des méthodes d'optimisation de ce type, on consultera (Hansen et Jaumard, 1997). Pour simplifier, nous admettrons qu'il existe trois familles de critères "naturels" en classification (Guénoche, 2003) :
- La séparation : une bonne partition présente des classes bien séparées ; on cherche à maximiser les écarts entre classes, qui sont fonctions des distances inter-classes.
- L'homogénéité : les classes sont les plus concises possible, on cherche à minimiser le diamètre, c'est-à-dire le maximum de distances intra-classes.
- La dispersion : on minimise une fonction d'inertie, la somme des écarts à un centre en norme quelconque, qu'il soit réel ou virtuel.

Généralement ces méthodes d'optimisation conduisent à des algorithmes NP-difficiles et le minimum atteint par une méthode de descente n'est pas optimal. On peut alors utiliser l'arsenal des méthodes d'optimisation stochastiques : la méthode de recherche Tabou (Glover et Laguna, 1997), le recuit simulé (Kirkpatrick *et col.*, 1983), ou les algorithmes génétiques (Holland, 1975). Ces heuristiques permettent de visiter une toute petite partie de l'ensemble des partitions à nombre de classes fixé et, à la fin on conserve la meilleure partition trouvée au cours de ces explorations.
Nous n'entrerons pas dans le détail de ces algorithmes, mais nous introduisons les méthodes de classification par densité dont nous avons découvert tardivement qu'elles remontaient à peu près à la même époque.

3.1.4 Classification par densité

Ces méthodes sont basées sur une idée très naturelle : considérer les objets dans leur voisinage, c'est-à-dire l'ensemble de leur plus proches voisins. Dans la première méthode (Wishart, 1969), la taille de ces voisinages est une constante fixée par l'utilisateur. Dans la seconde, dite "méthode de percolation" (Trémolières et Vanbaelinghem, 1977), un seuil de distance, également fixé par l'utilisateur, détermine le nombre d'objets considérés comme voisins, et qui n'est pas constant. Ce type de méthodes construit donc des voisinages pour définir des zones de forte densité constituant les classes ; elles dépendent de la définition de la densité. Ces deux méthodes partent de la matrice des distances entre objets.

Analyse modale de Wishart : La méthode de Wishart (1969) vise à enlever les objets isolés (considérés comme du "bruit") de manière à supprimer l'effet de chaînage entre classes et à détecter les objets denses du graphe. Cette méthode se base sur le calcul d'une densité pour chaque objet, définie comme la distance moyenne des K objets les plus rapprochés (où K est un paramètre fourni par l'utilisateur permettant de borner le voisinage de chaque objet). Parler ici d'une mesure de "densité" est un peu déplacé dans la mesure où un sommet ayant une faible valeur de densité sera d'autant plus proche de ses voisins et se situera donc dans une zone dense. Les objets sont considérés par valeur de "densité" croissante (puisque les plus denses ont les valeurs les plus faibles). Chaque objet est alors classé dans la ou les classes dont il est le plus proche (sa densité est inférieure à une distance entre cet objet et l'un des objets appartenant à une ou des classes). S'il est proche de plusieurs classes, ces classes seront fusionnées. S'il n'existe aucune classe proche, alors cet objet en initie une nouvelle.
Cette méthode ne classe pas les points isolés qui ont des "densités" très fortes, c'est l'élimination du "bruit". Le nombre de classes n'est pas à fixer, mais, encore une fois, il y a un paramètre à fixer : le nombre d'objets les plus proches.

Méthode de percolation de Trémolières : La méthode de Trémolières et Vanbaelinghem (1977) (puis (Trémolières, 1994)) permet, outre la détection des classes, la détection des objets isolés dans des zones non dense et des objets "frontières" – objets qui peuvent être rattachés à plusieurs classes. Les objets isolés et frontières restent non classés.
La méthode de percolation repose sur un paramètre fixé par l'utilisateur : la distance σ définissant l'étendue du voisinage $V(x_i, \sigma)$ d'un objet x_i.

Définition 1.5 *Le voisinage $V(x_i, \sigma)$ d'un objet $x_i \in O$, où d est la fonction de distance, est défini par :*

$$V(x_i, \sigma) = \{x_j \in O \text{ tel que } d(x_i, x_j) \leq \sigma\}$$

Les objets x_j qui sont à une distance inférieure ou égale à σ de l'objet x_i sont ses voisins. La densité de l'objet x_i est égale au nombre de ses voisins. Les objets sont ensuite placés dans une liste L et ordonnés par densité décroissante. Cinq ensembles d'objets sont alors créés et mis à jour à chaque itération :

- C est l'ensemble des objets déjà assignés à une classe.
- $\widetilde{C}$ est l'ensemble des objets voisins des objets appartenant à C :

$$\widetilde{C} = \{x_j \in O \setminus C \text{ tel que } \exists x_i \in C, d(x_i, x_j) \leq \sigma\}$$

- B est l'ensemble des points frontières :

$$B = \{x_i \text{ tels qu'il existe au moins deux classes } C_1 \text{ et } C_2 \text{ avec } V(x_i, \sigma) \cap C_1 \neq \{\emptyset\} \text{ et } V(x_i, \sigma) \cap C_2 \neq \{\emptyset\}\}$$

- F est l'ensemble des objets candidats à un regroupement :

$$F = O \setminus C \setminus B$$

– D est l'ensemble des voisins de l'objet le plus dense de F :

$$\text{Soit } x_i \in F \text{ l'élément de densité maximum }, D = V(x_i, \sigma)$$

Cet algorithme produit une partition et élimine les chaînes d'objets séparés par d'égales distances : selon le seuil de distance σ choisi, ils forment soit une classe unique, soit autant de classes qu'il existe d'objets. Il dépend encore une fois d'un paramètre fixé par l'utilisateur.

3.2 Classification par recherche de zones denses

Comme (Girvan et Newman, 2002), nous traitons des graphes et, comme (Wishart, 1969) et (Trémolières, 1994), nous recherchons des densités. Dans notre méthode ((Colombo *et col.*, 2003), (Colombo *et col.*, 2004) et (Guénoche, 2004)), la densité est évaluée en chaque sommet d'un graphe[37] à l'aide d'une fonction, puis nous recherchons des composantes connexes de forte densité. Nous travaillerons sur un graphe $\Gamma = (S, A)$ considéré comme connexe, où S est l'ensemble des sommets ($|S| = n$) et A est l'ensemble des arêtes ($|A| = m$). Le degré d'un sommet x sera noté $Dg(x) = |\{y \text{ tel que } (x, y) \in A\}|$ et le degré maximum du graphe sera noté δ. J'introduis maintenant quelques notations qui seront utilisées par la suite.
Soit Y une partie de S : on désigne par $\Gamma(Y)$ l'ensemble des sommets extérieurs à Y qui sont adjacents à des sommets de Y.

$$\Gamma(Y) = \{x \in S \setminus Y \text{ tel que } \exists y \in Y, (x, y) \in A\}$$

Le calcul du nombre de triangles est souvent utilisé. Nous distinguerons deux cas :

– Le nombre de triangles $N_t(x)$ dont x est un sommet :

$$N_t(x) = |\{(y, z) \in A \text{ tels que } (x, y) \in A, \text{ et } (x, z) \in A\}|$$

– Le nombre de triangles $N_T(x, y)$ dans lesquels l'arête (x, y) est impliquée :

$$N_T(x, y) = |\{x, y, z\} \in S \text{ tels que } (x, y), (x, z), (y, z) \in A|$$

Les différentes étapes de la méthode peuvent être résumées de la façon suivante :

– On calcule la densité en chaque sommet grâce à une fonction de densité De.
– On crée ensuite des *noyaux* qui sont des composantes connexes des sommets pour lesquels la densité est maximale localement et supérieure à la moyenne.
– Ensuite une étape d'extension permet de classer les éléments restants.

Je commencerai donc par présenter différentes fonctions de densité qui ont été étudiées.

3.2.1 Fonctions de densité locale

On évalue la densité en chaque sommet à l'aide d'une fonction de densité $De : S \mapsto \mathbb{R}_+$. Par définition, tous les sommets de degré 1 ont une densité égale à 0, ce qui évite d'avoir des valeurs indéfinies. Voici cinq fonctions dont l'influence sur la classification finale sera évaluée dans la partie "Résultats" :

[37] Tout comme dans les approches de Bader et Hogue (2003) ou encore Rougemont et Hingamp (2003).

– Il y a tout d'abord le simple degré rapporté au degré maximum :

$$De_1(x) = \frac{Dg(x)}{DgMax}$$

Mais cette fonction donne une place centrale aux sommets de fort degré, place qu'ils n'ont pas toujours.

– Le degré moyen dans le voisinage de x :

$$De_2(x) = \frac{Dg(x) + \sum_{y \in \Gamma(x)} Dg(y)}{(1 + Dg(x))}$$

Cette fonction compte de la même façon les arêtes qui sortent du voisinage de x que celles qui lient les sommets adjacents à x. La fonction De_3 pallie ce défaut.

– Le taux de triangles $Nt(x)$ passant par x.
Ce nombre est divisé par le maximum réalisable par un sommet de degré $Dg(x)$.

$$De_3(x) = \frac{2 \times N_t(x)}{Dg(x) \times (Dg(x) - 1)}$$

Cette fonction est la plus utilisée dans les approches similaires en classification à partir d'une instance (Rougemont et Hingamp, 2003). Un sommet dont tous les voisins sont adjacents deux à deux aura une densité maximale égale à 1. Par contre, cette fonction décroît très vite dès que les sommets adjacents à x ne sont pas tous connectés. Pour donner plus de poids aux sommets possédant beaucoup de connexions, nous introduisons la fonction De_4.

– Le pourcentage d'arêtes dans le voisinage de x, c'est-à-dire le nombre d'arêtes adjacentes à x plus celles formant triangle, le tout rapporté au nombre maximum d'arêtes du voisinage d'un sommet de degré $Dg(x)$.

$$De_4(x) = \frac{2 \times (Dg(x) + N_t(x))}{Dg(x) \times (Dg(x) + 1)}$$

Viennent ensuite deux fonctions de densité qui sont calculées à partir de distances.

– Densité d'après une distance d. Soit d_{max} la valeur de distance maximale. Connaissant la fonction de distance d, la fonction de densité se calcule par :

$$De_d(x) = 1 - \frac{\frac{\sum_{y \in S} d(x,y)}{Dg(x)}}{d_{max}}$$

La densité De_5 est calculée grâce à D_{e_d} en utilisant la distance de Czekanovski-Dice qui est une distance locale permettant d'exprimer une relation de voisinage. Pour une arête $(x, y) \in A$, la distance de Czekanovski-Dice est :

$$d_C(x, y) = 1 - \frac{2 \times N_T(x, y) + 2}{Dg(x) + Dg(y) + 2}$$

Deux points impliqués dans un grand nombre de triangles et ayant peu d'arêtes externes à ces triangles seront considérés comme proches.

Pour évaluer les deux premières fonctions de densité, il suffit de parcourir la liste des arêtes dont la longueur est bornée par $m \leq n\delta$. Pour les fonctions De_3 et De_4, il faut tester l'existence des arêtes dans le voisinage de x qui contient au plus δ sommets. Enfin, pour la fonction De_5, il faut tester l'existence de triangles pour chaque arête. La complexité du calcul de densité est donc en $O(n\delta)$ pour De_1 et De_2, en $O(n\delta^2)$ pour De_3 et De_4, et en $O(n\delta^3)$ pour De_5.

3.2.2 Hiérarchie de la densité

Les sommets du graphe peuvent être représentés par une classification hiérarchique. Pour un seuil de densité σ donné, on considère le graphe seuil Γ_σ de sommets S et dont les arêtes lient des sommets de densité supérieure ou égale à σ. Ses classes sont les parties connexes disjointes et, quand le seuil varie, toutes ces classes forment une hiérarchie. Je présente ici une méthode de construction directe de l'arbre de classification des sommets du graphe dans lequel chaque sous-arbre correspond à une partie connexe au seuil de densité.
A un seuil donné, deux classes sont nécessairement disjointes ; si elles avaient un élément commun, puisque ce sont des parties connexes, elles seraient connexes. Et, pour deux seuils $\sigma_1 < \sigma_2$, un noyau au seuil σ_2 est nécessairement inclus dans un noyau au seuil σ_1. Ainsi, pour toutes les valeurs de seuils, les noyaux sont des classes disjointes ou emboîtées. Elles vérifient presque l'axiomatique des hiérarchies puisque S est un noyau à la valeur minimale de densité ; par contre, tous les singletons n'ont pas la même valeur maximum de densité. Pour obtenir une hiérarchie, il suffit donc de donner aux singletons une densité égale à la plus forte valeur de densité $DeMax$. Cette hiérarchie est naturellement indicée par la densité, c'est pourquoi nous l'appelons *hiérarchie de la densité*. Une partie devient connexe à la densité minimum de ses éléments, ce qui définit un index. Pour respecter la décroissance de l'indice des classes suivant la relation d'inclusion, il suffit de complémenter les densités ; l'indice d'une classe Y est égal à $DeMax - min_{z \in Y} De(z)$ et l'indice de tous les singletons est fixé à 0. A cette hiérarchie ainsi indicée (Guénoche, 2004) correspond une distance ultramétrique, notée U_d, et donc un arbre de classification. C'est cet arbre que nous nous attachons à construire, et ce sans calculer U_d ni mesurer de distance sur S basée sur les arêtes de Γ.

Propriété 2.1 *Soit $L : A \mapsto \mathbb{R}$ définie par $L(x, y) = DeMax - min(De(x), De(y))$. Un arbre minimum de Γ valué par la fonction L est un arbre minimum de l'ultramétrique U_d.*

En effet, la distance $U_d(x, y)$ est l'indice de la classe de plus petit indice qui connecte x et y. Elle est égale à la longueur d'une plus longue arête d'un chemin entre x et y, pour lequel cette plus longue arête est de longueur minimum. Soit A_m un arbre minimum[38] de Γ valué par L. Sur le chemin de A_m entre x et y cette plus longue arête est de longueur minimum, par définition de A_m.

[38] Un arbre couvrant minimal est un arbre connectant tous les sommets du graphe, et tel que la somme des longueurs des arêtes soit minimale.

3.2.2.1 Du graphe Γ valué par L à un arbre minimum

L'algorithme présenté ci-dessous est la restriction au graphe Γ valué par L de l'algorithme de Prim (1957) pour la construction d'un arbre minimum d'une dissimilarité. Au fil des itérations, on maintient à jour une structure de données qui permet de savoir :

- si un sommet x est dans l'arbre minimum (on notera $A_m(x) = 1$) ou s'il est encore hors arbre ($A_m(x) = 0$), et
- pour tout sommet hors arbre, quelle est sa distance à l'arbre, c'est à dire la longueur minimum d'une arête du graphe qui permettrait de le connecter (s'il en existe).

Pour un sommet hors arbre x, on note $Adj(x)$ le sommet dans l'arbre tel que $L(x, Adj(x))$ est de longueur minimum. Définissons $Dis(x) = L(x, Adj(x))$. Si x n'a aucun sommet adjacent dans l'arbre, on pose $Dis(x) = DeMax$.

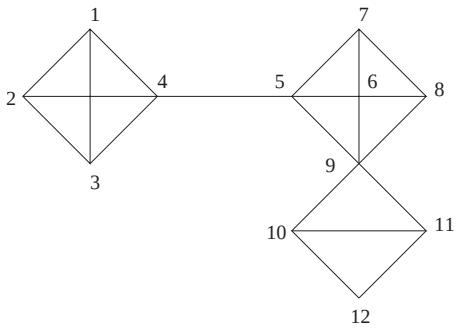
Initialement, on part du premier sommet dans le graphe (numéroté 1). On fixe à $L(1, x)$ la distance de tout sommet x adjacent à 1 et à $DeMax$ la distance de ceux qui ne sont pas adjacents. Le sommet 1 est considéré comme placé dans l'arbre minimum.

```
Am(1) <- 1;
Pour x <- 2 à n
    Am(x) <- 0;
    Si (x est adjacent à 1)
        Dis(x) <- L(1,x);
        Adj(x) <- 1;
    Sinon
        Dis(x) <- DeMax;
```

A chaque itération, on place dans l'arbre un nouveau sommet, noté *piv* ; il s'agit de celui qui est à distance minimum de l'arbre. Puis, pour tous les sommets x adjacents à *piv* qui ont une distance à l'arbre supérieure à la longueur de l'arête (piv, x), on met à jour cette distance et *piv* devient leur plus proche voisin dans l'arbre.

```
DisMax <- DeMax;
Pour x <- 2 à n
    Si (Am(x)=0 et Dis(x)<=DisMax)
        DisMax <- Dis(x);
        piv <- x;
Am(piv) <- 1;
Pour tout sommet x adjacent à piv
    Si (Am(x)=0 et L(piv,x)<Dis(x))
        Dis(x) <- L(piv,x);
        Adj(x) <- piv;
```

Après $n-1$ itérations, la liste des arêtes est $(x, Adj(x))$ de longueur $L(x)$, pour x variant de 2 à n.



S	1	2	3	4	5	6	7	8	9	10	11	12
De_4	**1.0**	**1.0**	**1.0**	0.7	0.6	0.8	0.833	0.833	0.533	0.833	0.833	**1.0**

FIG. 3.25 – Graphe avec $n = 12$ et $m = 20$; la table indique la densité de chaque sommet calculée à l'aide de la fonction De_4. La valeur de densité maximale est $DeMax = 1.0$.

Exemple : En se basant sur le graphe de la figure 3.25 où la densité est calulée par la fonction De_4 et a pour valeur maximale $DeMax = 1.0$, on obtient par exemple : $L(6,5) = 1.0 - Min(De_4(6) = 0.8, De_4(5) = 0.6) = 1.0 - 0.6 = 0.4$. En effectuant les autres calculs, l'arbre minimum de la hiérarchie de la densité est alors :

```
 Arêtes : Longueur
 2 -  1 : 0.000      3 -  1 : 0.000      4 -  1 : 0.300      5 -  4 : 0.400
 6 -  5 : 0.400      7 -  6 : 0.200      8 -  7 : 0.167      9 -  5 : 0.477
10 -  9 : 0.477     11 - 10 : 0.167     12 - 11 : 0.167
```

3.2.2.2 De l'arbre minimum au dendrogramme

A partir de cet arbre minimum valué par L, on pourrait construire l'ultramétrique associée. Si on note $[x, y]$ le chemin dans l'arbre entre x et y, elle est définie par :

$$U_d(x, y) = max_{(u,v)\in[x,y]} L(u, v)$$

Puis, par un algorithme de classification hiérarchique, on déterminerait le dendrogramme correspondant. Les deux parties de cet algorithme sont en $O(n^2)$, mais il nécessite de stocker la matrice des distances ultramétriques et de la mettre à jour à chaque itération. C'est pourquoi nous préférons calculer directement la structure et l'arbre de classification. Il est stocké dans deux tableaux $Tree$ et $Long$, initialisés à 0, et indexés sur les $2n - 2$ sommets ; les n premiers correspondent aux éléments de S, et les $n - 2$ suivants aux classes propres de la hiérarchie.

On considère les arêtes de A_m dans l'ordre des longueurs croissantes. Soit (x, y) l'arête courante de longueur $L(x, y)$ et n_s le nombre de sommets courant dans l'arbre de classification, initialement fixé à n. Pour chaque arête on applique la procédure ci-dessous. Partant de x, puis de y, on

remonte vers la racine de l'arbre de classification, jusqu'à la plus grande classe de la hiérarchie, au seuil précédent $L(x, y)$, qui contient cet élément. Ce sont ces deux classes qui sont réunies dans une nouvelle classe numérotée n_s.

```
ns <- ns + 1;
u <- x;
SomL <- 0;
Tant que (Tree(u)=0) Faire
    u <- Tree(u);
    SomL <- SomL + Long(u, Tree(u));
Tree(u) <- ns;
Long(u) <- De(x) - SomL;
u <- y;
SomL <- 0;
Tant que (Tree(u)=0) Faire
    u <- Tree(u);
    SomL <- SomL + Long(u, Tree(u));
Tree(u) <- ns;
Long(u) <- De(y) - SomL;
```

Après $n-1$ itérations, la liste des arêtes de l'arbre de classification est $(x, Tree(x))$ de longueur $Long(x)$, pour x variant de 1 à $n-2$. La racine de l'arbre a pour numéro $n_s = 2n-1$. L'arbre hiérarchique correspondant aux données de l'exemple traité est représenté dans la figure 3.26. Ce type de représentation est utile lorsque l'on veut apprécier le nombre de classes obtenues en assignant une valeur à σ[39]. Toutefois, lorsque l'on désire une méthode totalement automatique et que l'on ne connaît pas le nombre de classes attendu, cette représentation n'est pas satisfaisante. C'est pourquoi nous avons développé un algorithme ne nécessitant aucun paramètre pour déterminer une partition.

3.2.3 Algorithme de partitionnement

L'algorithme se déroule en deux phases : la construction des noyaux des classes, puis la complétion de ces noyaux avec tout ou partie des éléments restant.

3.2.3.1 Construction des noyaux des classes

Les classes recherchées sont, par définition, des parties connexes du graphe Γ. Ces classes correspondent à des valeurs de densité considérées comme fortes. Notre idée initiale était de chercher un seuil de densité et de considérer le sous-graphe partiel dont les sommets ont une densité supérieure à ce seuil ; les classes seraient alors ces composantes connexes. Cette méthode ne donne pas de très bons résultats pour deux raisons :

– Le choix du seuil est un problème délicat ; au seuil maximum, s'il est unique, il n'y a qu'une classe à un seul élément. Au seuil 0, il n'y a qu'une classe qui contient tous les éléments.

[39] La variation de ce seuil fait évoluer le nombre de classes.

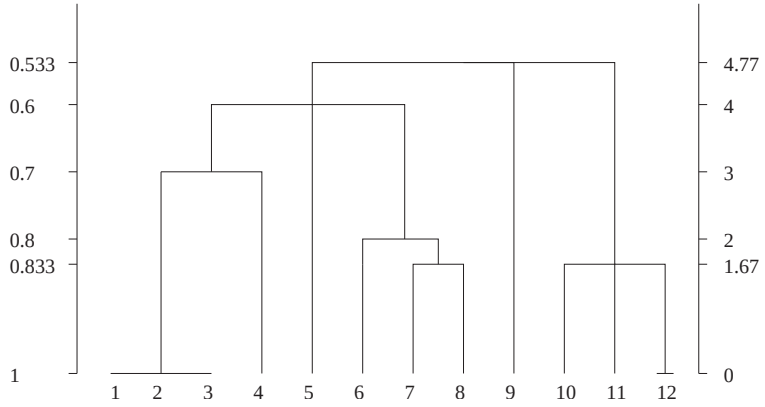


FIG. 3.26 – Arbre hiérarchique de la densité De_4 du graphe de la figure 3.25 encadré entre les échelles de la densité et de l'indice de la hiérarchie.

Une valeur par défaut, égale à la densité moyenne du graphe, ne donne pas toujours le nombre de classes attendues[40].

– En énumérant tous les seuils possibles, nous nous sommes aperçus qu'aucun seuil n'était satisfaisant, en particulier pour les fonctions De_1 et De_2. En faisant décroître le seuil, on n'obtient bien souvent qu'une seule classe. Pour les fonctions De_3 à De_5, il y a beaucoup de fluctuations dans les fortes valeurs, donc plusieurs classes, mais elles contiennent très peu d'éléments.

Puisqu'il n'y a pas de seuil global, valable pour l'ensemble des classes, nous avons décidé de considérer les maxima locaux de la densité pour construire les noyaux des classes.
Un noyau, noté N, est une partie de S connexe dans Γ. On commence par rechercher tous les maxima locaux de la fonction de densité et on considère le sous-graphe partiel de Γ réduit à ces sommets.

$$\forall x \in N, \forall y \in \Gamma(x) \text{ on a } De(x) \geq De(y).$$

Les noyaux initiaux sont les composantes connexes de ce graphe. En d'autres termes, si plusieurs maxima locaux sont adjacents, ils ont même valeur et sont alors réunis ; sinon les noyaux initiaux sont des singletons. Ensuite, on affecte à chaque noyau N les sommets de $\Gamma(N)$ qui ne sont adjacents qu'à un seul noyau, à condition qu'il ait une densité supérieure ou égale à la densité moyenne. On évite ainsi toute ambiguïté dans l'affectation ou toute attribution à une seule classe quand plusieurs sont possibles (l'indécision étant alors conservée). La composition des noyaux est irréversible : leur nombre définit le nombre de classes, qui ne sera pas modifié par la suite.
Nous essayons ensuite d'étendre les noyaux ainsi constitués en ajoutant d'autres éléments.

[40] Voir la partie "Validation par simulations" p. 80.

3.2.3.2 Extension des noyaux en classes denses

Nous avons implémenté les deux stratégies suivantes :

Maximiser le degré moyen : Le principe de cette extension est d'ajouter un à un au noyau tous les éléments qui s'y rattachent et qui permettent d'augmenter son degré moyen. Soit C une classe initialement égale à un noyau N ;

- on calcule le degré moyen de la classe, en ne considérant que les arêtes internes,
- le nombre maximum de connexions entre la classe et un élément de $\Gamma(C)$;
- si ce nombre est supérieur ou égal au degré moyen, on ajoute à la classe les éléments qui possèdent un nombre maximum de connexions.

Si au moins un élément a été ajouté, on réitère la procédure.

Ainsi, les classes sont toujours des parties connexes. La valeur calculée de la densité a été utilisée pour initialiser ces classes qui s'enrichissent couche par couche. Selon cette procédure, un élément peut être ajouté à plusieurs noyaux et donc les classes denses ainsi réalisées ne sont pas nécessairement disjointes.

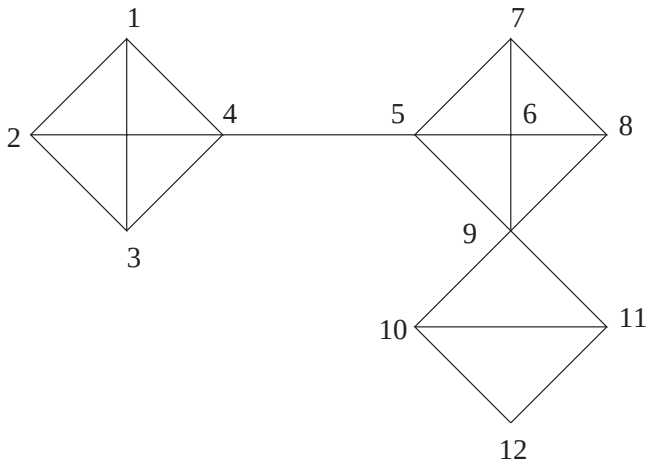
Exemple : Les noyaux du graphe de la figure 3.27 ont été calculés à partir des maxima locaux de la densité De_4 (voir la représentation en trois dimensions des densités sur le graphe en figure 3.28), soit $\{1, 2, 3\}$, $\{7, 8\}$ et $\{12\}$. La valeur moyenne de densité est 0.83. Seuls les sommets 10 et 11 peuvent compléter le troisième noyau ; ils sont édités en amont du signe +. Pour les classes étendues, le sommet 4 se rattache à la première classe ; pour la deuxième, le sommet 6 permet d'élever le degré moyen puis les sommets 5 et 9, qui ont deux connections vers $\{6, 7, 8\}$, s'y rattachent aussi. Pour la classe 3, le sommet 9 s'y rattache également. Au delà de la flèche figure, entre parenthèses, le degré interne moyen de la classe résultante :

```
Classe 1 :  1  2  3 +     4 -> (3.0)
Classe 2 :     7  8 + 6 5 9 -> (3.2)
Classe 3 : 10 11 12 +     9 -> (2.5)
```

On remarque que le sommet 9 a été ajouté aux deux classes 2 et 3. Avec cette stratégie, outre le fait de pouvoir classer un même élément dans plusieurs classes, on a pu s'apercevoir (cf. "Validation par simulations" p. 80) qu'elle avait tendance à classer un quart des éléments dans les noyaux et la moitié dans les classes étendues. Elle ne classe donc pas forcément tous les sommets du graphe. Pour cela, nous avons développé une seconde méthode d'extension des noyaux.

Maximiser le nombre d'éléments classés : On considère qu'à l'issue de la première étape, on a p noyaux notés N_i. Soit $L = S - \bigcup_{1 \leq i \leq p} N_i$ l'ensemble des q sommets restant à classer. On va traiter ces sommets séquentiellement en les rattachant aux noyaux auxquels ils sont principalement liés.
Pour chaque sommet x de L pris dans l'ordre de densité décroissante :

- pour chaque noyau N_i, on calcule le nombre c_i de ses connexions à x et s_i, le nombre



S	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Dg	3	3	3	4	4	4	3	3	5	3	3	2
De_1	0.6	0.6	0.6	0.8	0.8	0.8	0.6	0.6	1.0	0.6	0.6	0.4
De_2	0.812	0.812	0.812	0.85	1.0	0.95	0.875	0.937	0.916	0.812	0.812	0.667
De_3	1.0	1.0	1.0	0.5	0.333	0.667	0.667	0.667	0.3	0.667	0.667	1.0
De_4	1.0	1.0	1.0	0.7	0.6	0.8	0.833	0.833	0.533	0.833	0.833	1.0
De_5	0.963	0.963	0.963	0.767	0.603	0.826	0.769	0.746	0.615	0.819	0.819	0.857

FIG. 3.27 – Graphe avec $n = 12$ et $m = 20$; la table indique le degré et la densité de chaque sommet. Pour pouvoir comparer les différentes valeur de densité, les résultats de la fonction De_2 ont été normalisés.

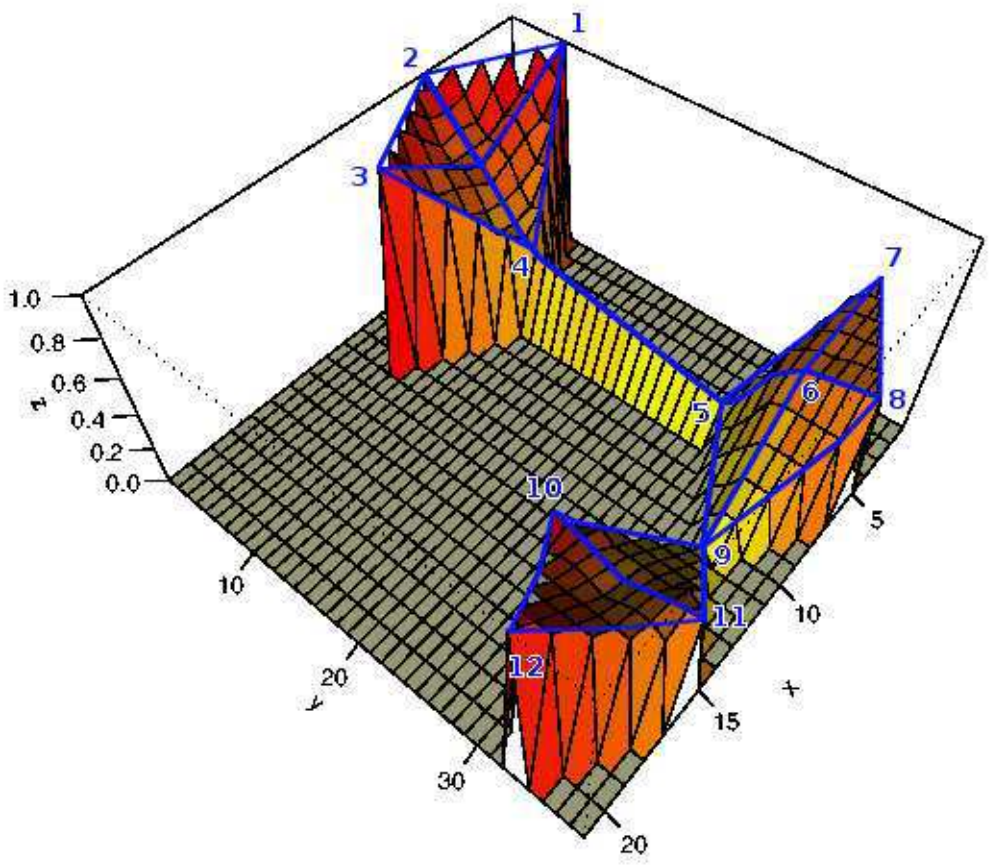


FIG. 3.28 – Représentation en trois dimensions des densités calculées par la fonction De_4 sur le graphe de la figure 3.27. Cette représentation a été obtenue en utilisant la librairie BioGraph (Colombo, 2004) et le language R (Ihaka et Gentleman, 1996).

d'éléments du noyau N_i :

$$c_i = |\Gamma(x) \bigcap N_i| \text{ et } s_i = |x \in N_i|$$

- x est connecté au noyau N_j tel que c_j est maximum et, en cas d'égalité entre plusieurs noyaux, celui de s_j minimal ;
- les quantités c_i et s_i sont mises à jour.

On assure ainsi que chaque élément sera affecté à un seul noyau ; la décision en cas d'égalité revient à placer x dans la classe ayant le plus petit nombre d'arêtes internes, ce qui en augmente le taux. Cette règle conduit également à équilibrer les classes, ce qui peut se justifier en fonction du domaine d'application.

Exemple : Reprenons le graphe de la figure 3.27 et ses noyaux $\{1, 2, 3\}$, $\{7, 8\}$, et $\{10, 11, 12\}$. On obtient :

```
L = (10, 11, 6, 5, 4, 9)
Classe 1 :  1  2  3 + 4
Classe 2 :     7  8 + 6 5 9
Classe 3 : 10 11 12 +
```

On remarque que le sommet 9, au moment où il est classé, a trois connexions avec la classe 2 et deux connexions avec la classe 3. Il est donc affecté à la classe 2.

3.2.3.3 Complexité

Posons : n le nombre de sommets, m le nombre d'arêtes, p le nombre de noyaux, q le nombre de sommets non-classés dans les noyaux avant extension, et δ le degré maximum du graphe.
La procédure de construction des noyaux est en $O(m) \approx O(n\delta)$. Pour l'étape d'extension, on commence par calculer le degré moyen de chaque noyau et le nombre de connexions des éléments adjacents ; cette partie est en $O((n-q)\delta)$.

- Dans le premier cas, à chaque itération, on retient les éléments dont le degré est suffisant et on met à jour le degré moyen de la classe et les degrés des autres éléments soit $O(pq\delta)$; on remarque que le nombre d'itérations est borné par δ, soit une complexité en $O(pq\delta^2)$;
- dans le second cas, on affecte à chaque itération un seul élément hors noyau, et on met à jour les p valeurs c_i et s_i en examinant au plus δ arêtes, soit en $O(pq\delta)$ pour toutes les classes.

En remarquant que $q < n$ et que l'étape initiale ne compte qu'une fois par rapport à p, la compexité de la procédure d'extension est donc bornée par $O(np\delta^2)$.

Par son très faible coût en temps de calcul, cette méthode permet de traiter de gros graphes de façon très efficace. Ceci est très important pour les données biologiques en constante augmentation : en un an on est passé de 60 génomes bactériens entièrement séquencés à plus d'une centaine et chacun possède plusieurs milliers de gènes.

3.2.4 Validation de la méthode par simulations

La validation peut être effectuée sur des graphes aléatoires. Mais ces résultats, prouvant le bien fondé d'une méthode de partitionnement, seront liés à la qualité du générateur implémenté. De plus, si les différents paramètres sont correctement choisis (nombre de sommets, degré de distribution, ...), de tels graphes peuvent représenter de manière fidèle des problèmes réels ((Newman *et col.*, 2001), (Newman, 2003)).
Les graphes considérés ici sont des graphes dont le degré des sommets n'est pas fixé a priori.
De plus, nous devons être capable de comparer les résultats obtenus par différentes méthodes de partitionnement. Pour celà nous aurons recours à des critères d'évaluation.

3.2.4.1 Les critères d'évaluation : mesure de la qualité d'une partition

Il existe de nombreux critères permettant de mesurer la qualité (interne ou externe) d'une partition. Ces mesures dépendent une fois de plus des données étudiées. Mais on peut introduire d'autres critères, notamment si l'on connait la partition initiale[41].
Pour chaque classe calculée on peut, par exemple, rechercher quelle est la classe initiale dont elle se rapproche le plus. On calcule alors le pourcentage d'éléments communs.

Définition 2.2 *Soit n' le nombre d'éléments classés dans p' classes $C'_1, ..., C'_{p'}$ constituant une partition P'. La partition initiale P en p classes est connue. On évalue les classes de P' par rapport à celles de P par : $n_{i,j} = |C_i \cap C'_j|$ (voir table 3.8). La classe majoritaire correspondant à C'_j est notée $\Theta(C'_j)$. Il s'agit de la classe de P qui contient le plus d'éléments de C'_j. $\Theta(C'_j) = C_k$ si et seulement si $\forall 1 \leq i \leq p, n_{k,i} \geq n_{i,j}$. Le pourcentage d'éléments de l'une des classes calculées qui appartient à la classe majoritaire correspondante dans la partition initiale est alors :*

$$\tau_e = \frac{\sum_i |\Theta(C'_i) \cap C'_i|}{n'}$$

On peut également déterminer pour chaque paire d'éléments d'une même classe s'ils apparaissent également en paire dans une des classes de la partition initiale.

Définition 2.3 *τ_p : le pourcentage de paires d'éléments d'une même classe qui sont dans une même classe de la partition initiale.*

$$\tau_p = \forall x, y \sum_{i=1}^{p} \sum_{j=1}^{p'} \frac{|\{(x,y) \in (C_i \times C_i) \cap (C'_j \times C'_j)\}|}{\frac{1}{2} \sum_j |C'_j \times (C'_j - 1)|}$$

[41] En marge de ces critères d'évaluation, Watts et Strogatz (1998) ont développé un coefficient de clustering permettant de mesurer le "degré de clustering" d'un graphe. Ce coefficient peut s'écrire $C = \frac{3\times \text{ nombre de triangles du graphe}}{\text{nombre de triplets de sommets connectés}}$ où un "triplet de sommets connectés" est un groupe de trois sommets dans lequel au moins un est connecté aux deux autres.

$\bigcap$	C'_1	$\cdots$	$C'_{p'}$
C_1		$\vdots$	
$\vdots$	$\cdots$	$n_{i,j}$	$\cdots$
C_p		$\vdots$	

TAB. 3.8 – Calcul de $n_{i,j}$ permettant de déterminer la classe majoritaire correspondant à C'_j.

Ce critère est classique dans la comparaison de partitions : on compte les paires d'éléments sur lesquelles les deux partitions P et P' sont en accord ou en désaccord. τ_p n'est qu'un des quatre critères possibles :

- N_{11} : le nombre de paires d'éléments réunies dans C et dans C'– les paires en accord. Il s'agit du numérateur de τ_p.
- N_{00} : le nombre de paires d'éléments séparées dans C et C'.
- N_{10} : le nombre de paires d'éléments qui sont dans une même classe dans C mais dans des classes différentes dans C'.
- N_{01} : le nombre de paires d'éléments qui sont dans une même classe dans C' mais dans des classes différentes dans C.

Les quatres critères vérifient : $N_{11} + N_{00} + N_{10} + N_{01} = \frac{n \times (n-1)}{2}$.

Définition 2.4 *De ces critères nous pouvons définir un indice très couramment utilisé (et modifié) : l'indice de Rand (1971).*

$$\tau_R = \frac{N_{11} + N_{00}}{\frac{n \times (n-1)}{2}}$$

Il s'agit d'un coefficient d'accord sur les deux partitions considérées comme équivalentes.

On peut alors compter le nombre de paires trouvées par rapport au nombre de paires initiales et modifier ce critère par :

$$\tau_A = \frac{N_{11}}{\frac{1}{2} \sum\limits_{i=1,\ldots p} C_i \times (C_i - 1)}$$

Enfin, certaines méthodes peuvent ne pas classer tous les éléments, il est alors intéressant de connaître la proportion d'éléments qui ont été classés. Il s'agit plus ici d'évaluer l'efficacité d'une méthode de classification que la partition trouvée.

Définition 2.5 *τ_c : le pourcentage d'éléments classés. Soit n le nombre d'éléments total et n' le nombre d'éléments classés : $\tau_c = \frac{n'}{n}$.*

Remarque : Dans le cas de τ_p et de τ_e, la valeur maximale peut être atteinte alors que les partitions initiales et finales sont différentes. Ceci se produit lors d'un "sur-découpage" de la partition finale par rapport à la partition initiale. Par contre τ_R et τ_A auront leur valeur maximale uniquement lorsque les deux partitions seront identiques.

3.2.4.2 Les graphes aléatoires

Un graphe aléatoire est un ensemble d'arêtes connectant des sommets deux à deux de manière aléatoire. On considère que la présence ou l'absence d'une arête entre deux sommets est indépendante de la présence ou l'absence d'une autre arête. Ainsi, chaque arête peut être considérée comme étant présente avec une probabilité indépendante p. On considère également que le graphe ne comporte qu'une seule composante connexe : l'étude d'une méthode de partitionnement sur un graphe possédant des composantes connexes serait biaisée, ces dernières définissant un pré-découpage – et donc une simplification – du problème.

Générateur aléatoire : Pour tester notre approche, il faut disposer de graphes dans lesquels il y a des classes de densité supérieure à la moyenne. Le générateur de graphes aléatoires dépendra alors de quatre paramètres :

- N : le nombre de sommets du graphe,
- q : le nombre de classes désiré,
- p_i : la probabilité d'apparition d'une arête interne,
- p_e : la probabilité d'apparition d'une arête externe.

La densité[42] interne à chaque classe et la densité externe sont des valeurs correspondant aux probabilités d'apparitions des arêtes p_i et p_e.
On peut définir les densité interne et externe comme :

Définition 2.6 *Soit* $\Gamma_1 = \Gamma/C_1 = (C_1, A \cap (C_1 \times C_1)) = (S_1, A_1)$*, où* $|S_1| = n_1$*, un sous-graphe de* Γ *définissant une classe. La densité interne de la classe* Γ_1 *sera alors :*

$$p_i = \frac{2 \times |A_1|}{n_1 \times (n_1 - 1)}$$

Il s'agit du rapport de la somme des arêtes de Γ_1 sur le nombre d'arêtes maximum réalisable avec les sommets de S_1.

Définition 2.7 *Soient* Γ_k*,* k *sous-graphes de* Γ *définissant* k *classes. La densité externe de* Γ *est alors :*

$$\forall (x,y)_k \in A_k \times A_k, p_e = \frac{\sum_{i \neq j} (x,y) \text{ tel que } x \in A_i \text{ et } y \in A_j}{\sum_{i \neq j} (\sum_{\Gamma_i} (x,y)_i \times \sum_{\Gamma_j} (x,y)_j)}$$

La densité externe est définie comme le rapport du nombre d'arêtes inter-classes sur le nombre d'arêtes inter-classes maximum réalisable.

[42] Une densité s'exprime comme une valeur positive comprise entre 0 – peu dense – et 1.0 – très dense.

Pour construire un graphe, on commence par tirer aléatoirement (suivant une distribution aléatoire uniforme[43]) une partition des N éléments en q classes notées $C_1, ..., C_q$. Puis, pour chaque paire d'éléments (x, y), on tire un nombre r au hasard ($0 \leq r \leq 1$) et l'on ajoute éventuellement l'arête correspondante :

- si $(x, y) \in C_i \times C_i$ et $r \leq p_i$,
- si $(x, y) \in C_i \times C_j$ avec $i \neq j$ et $r \leq p_e$.

Cette procédure ne garantit nullement des graphes ayant précisément q classes denses : elles peuvent se décomposer ou se mélanger en fonction des arêtes tirées au hasard. De même, les densités réelles ne sont pas nécessairement égales à p_i et p_e mais en pratique on s'en écarte peu et donc, en moyenne, ce sont bien ces paramètres que l'on retrouve.
En prenant un nombre de sommets suffisamment important et une densité externe suffisamment élevée, on minimise le risque d'obtenir plusieurs composantes connexes. Toutefois, pour palier cette éventualité, une fois le graphe généré on vérifie qu'il n'y ait bien qu'une seule composante connexe. Si tel n'est pas le cas, le graphe est rejeté et doit être à nouveau généré[44].
Ce générateur pourrait être amélioré en tenant compte d'une remarque évoquée dans (Barabási et Albert, 1999) : généralement la distribution des degrés des sommets de graphes aléatoires suit une distribution binomiale, ce qui n'est pas le cas des graphes basés sur des problèmes réels. Pour éviter cette distribution, on pourrait par exemple choisir d'affecter une densité interne différente à chaque classe du graphe.

Résultats sur les graphes aléatoires : Pour valider notre approche, nous avons appliqué notre algorithme sur des graphes générés de manière aléatoire. Les résultats sont des moyennes obtenues sur 200 graphes de 100 sommets répartis en 3 classes, avec une densité interne $p_i = 0.5$ et une densité externe $p_e = 0.1$. Ces chiffres semblent définir un problème facile, mais il suffit de considérer que si les q classes ont le même nombre d'éléments, il y a $p_i \times \frac{n(n-q)}{2q}$ arêtes intra-classes et $p_e \times \frac{n^2(q-1)}{2q}$ arêtes inter-classes. Il y a donc en moyenne 1216 arêtes dans nos graphes, ce qui fait une densité moyenne de 0.245, soit près de la moitié de ce que l'on trouve dans les classes.
La table 3.9 correspond en ligne aux trois types de classes calculées et en colonnes aux fonctions de densité. Chaque case de la table contient les valeurs τ_e, τ_p, et τ_c (voir pp 77-79). Plus les valeurs sont fortes, plus la fonction de densité est efficace. Les trois dernières lignes donnent : l'indice de Rand τ_R, l'indice de Rand modifié τ_A, et le nombre moyen de classes obtenues.

La supériorité des fonctions De_3 à De_5 est évidente. Elle est due au fait que l'on comptabilise les arêtes internes à la classe, et que l'on trouve un nombre de noyaux satisfaisant (avec toutefois un léger avantage pour la fonction De_5). La fonction De_1 est trop peu discriminante pour générer suffisamment de maxima locaux. Comme ce sont eux qui conditionnent le nombre de classes, celles-ci sont mal définies par rapport aux classes initiales.
Les performances des fonctions De_3 à De_5 sont satisfaisantes, que ce soit au niveau des noyaux, des classes étendues ou des partitions. En moyenne 25% des éléments sont classés dans les noyaux et 50% dans les classes étendues. Plus de 95% des éléments réunis dans les uns et les

[43] *rand()* du langage C

[44] Une version de ce générateur est disponible sur le site du CPAN dans une librairie Perl (Colombo, 2004) : http ://www.cpan.org/∼baldr/BioGraph.

	De_1			De_2			De_3		
	τ_e	τ_p	τ_c	τ_e	τ_p	τ_c	τ_e	τ_p	τ_c
Noyaux	.79	.66	.26	.87	.78	.22	.96	.95	.27
Extension 1	.85	.76	.43	.90	.83	.39	.98	.96	.47
Extension 2	.72	.63	1.00	.70	.61	1.00	.94	.92	1.00
Indice τ_R	.71			.68			.87		
Indice τ_A	.74			.72			.90		
Nb. de classes	2.46			2.46			3.38		

	De_4			De_5		
	τ_e	τ_p	τ_c	τ_e	τ_p	τ_c
Noyaux	.96	.93	.25	.96	.94	.31
Extension 1	.97	.96	.50	.97	.95	.44
Extension 2	.95	.93	1.00	.88	.84	1.00
Indice τ_R	.86			.87		
Indice τ_A	.90			.91		
Nb. de classes	5.58			3.06		

TAB. 3.9 – Résultats moyens obtenus sur 200 graphes aléatoires de 100 sommets répartis en 3 classes. Les paramètres utilisés sont : $p_i = 0.5$ et $p_e = 0.1$. "Extension 1" indique les résultat obtenus en utilisant la stratégie "Maximiser le degré moyen" et "Extension 2" ceux de la stratégie "Maximiser le nombre d'éléments classés".

autres proviennent bien des classes denses initiales. Pour les classes totales, en moyenne, 90% des affectations sont correctes. Les résultats obtenus grâce à ces trois fonctions sont semblables. Le seul critère pouvant les différencier est le nombre moyen de classes obtenus : on note un léger avantage pour la fonction De_5.
Par la suite, je me suis intéressé au générateur aléatoire de Girvan et Newman (2002) que j'ai codé et utilisé pour pouvoir comparer les résultats obtenus par ma méthode et par la méthode développée par Newman (2004).

Générateur aléatoire de Girvan et Newman : Dans leur article sur les structures en communauté, Girvan et Newman (2002) utilisent des graphes aléatoires générés de la manière suivante :
Chaque graphe possède $n = 128$ sommets et chacun d'eux est connecté à exactement $z = 16$ autres sommets. Les sommets sont divisés en $C = 4$ classes : pour chaque sommet d'une classe C_i, il possède z_{in} arêtes internes vers des sommets de C_i et $z_{out} = z - z_{in}$ arêtes externes vers des sommets de C_j avec $j \neq i$. Ces arêtes sont, bien sûr, tirées de manière aléatoire. D'après les définitions 2.6 et 2.7 sur les densités dans un graphe, on en déduit donc :

$$p_e = \frac{\frac{z_{out} \times N}{2}}{\frac{4\times(4-1)}{2} \times 32 \times 32} = \frac{z_{out}}{96}$$

$$p_i = \frac{\frac{z_{in} \times 32}{2}}{\frac{32\times 31}{2}} = \frac{z_{in}}{31} = \frac{z - z_{out}}{31}$$

On ne parvient pas toujours à obtenir de tels graphes. Si le graphe en cours de génération brise une contrainte, il est rejeté et l'on réitère la procédure.
Dans leur étude, Girvan et Newman (2002) font varier z_{out} de 1 à 8 ce qui, au niveau des densités, implique des variations de $(p_i = 0.48, p_e = 0.01)$ à $(p_i = 0.25, p_e = 0.08)$. On s'aperçoit donc que les communautés sont toujours relativement bien isolées quelle que soit la valeur de z_{out}. Pour des valeurs de z_{out} faibles, il y a de fortes probabilités d'obtenir plusieurs composantes connexes. De plus, cette méthode génère une classe de graphes beaucoup trop contrainte[45] et ne peut être utilisée pour valider une méthode appliquée aux données biologiques que nous étudions. Une méthode validée sur de tels graphes pourrait ne résoudre des problèmes que sur des graphes possédant cette structure très particulière (classes parfaitement équilibrées, de densité interne identique, et de degré constant). Toutefois, pour pouvoir comparer notre méthode à celle de Girvan et Newman (2002), nous avons quand même étudié les résultats produits par ce générateur.

Résultats sur les graphes aléatoires de Girvan et Newman : J'ai ensuite testé notre méthode en l'appliquant aux graphes générés par la méthode de Girvan et Newman (2002). Ceci permet de vérifier la quasi-équivalence des résultats obtenus en appliquant notre algorithme aux graphes des deux générateurs en prenant des paramètres équivalents ($z_{in} = 15$ et $(p_i, p_e) = (0.48, 0.01)$ pour la table 3.10, et $z_{in} = 8$ et $(p_i, p_e) = (0.25, 0.08)$ pour la table 3.11). On s'aperçoit tout de même que la structure imposée au graphe par le générateur de Girvan et

[45] Nombre de classes identiques, possédant le même nombre d'éléments, et intrinsèquement de degré fixé (Milo *et col.*, 2004).

	Graphes aléatoires			**Girvan et Newman**		
	τ_e	τ_p	τ_c	τ_e	τ_p	τ_c
Noyaux	1.0	1.0	.37	.95	.86	.31
Extension 1	1.0	1.0	.48	.96	.89	.53
Extension 2	1.0	1.0	1.0	.99	.90	1.0
Indice τ_R	.97			.91		
Indice τ_A	.98			.93		
Nb. de classes	4.91			5.68		

TAB. 3.10 – Résultats moyens obtenus sur 200 graphes aléatoires de 128 sommets répartis en 4 classes avec la fonction de densité De_5. Les paramètres utilisés sont : $p_i = 0.48$ et $p_e = 0.01$ pour les graphes aléatoires, et $z_{in} = 15$ pour les graphes aléatoires de Girvan et Newman.

	Graphes aléatoires			**Girvan et Newman**		
	τ_e	τ_p	τ_c	τ_e	τ_p	τ_c
Noyaux	.8	.64	.26	.68	.41	.26
Extension 1	.78	.64	.38	.68	.44	.39
Extension 2	.63	.47	1.0	.6	.41	1.0
Indice τ_R	.74			.7		
Indice τ_A	.77			.73		
Nb. de classes	6.82			7.11		

TAB. 3.11 – Résultats moyens obtenus sur 200 graphes aléatoires de 128 sommets répartis en 4 classes avec la fonction de densité De_5. Les paramètres utilisés sont : $p_i = 0.25$ et $p_e = 0.08$ pour les graphes aléatoires, et $z_{in} = 8$ pour les graphes aléatoires de Girvan et Newman.

Newman semble poser plus de problèmes : les résultats sont légèrement moins bons. De plus, alors que la méthode de Newman s'effondre pour des valeurs de z_{in} inférieures à 10 avec moins de 50% d'arêtes correctement classées (figure 3.29), notre méthode semble plus robuste et l'on obtient des résultats qui restent corrects.

3.3 Application aux transporteurs ABC

Dans cette partie, je présenterai les différents résultats obtenus lors de l'étude d'un cas réel avec l'application à une famille de transporteurs ABC.

3.3.1 Les familles de transporteurs dans ABCdb

Les transporteurs de la base ABCdb ont tout d'abord été organisés en sous-familles chez *Bacillus subtilis* (Quentin *et col.*, 1999) :

- par la construction d'un arbre à partir des séquences NBD en utilisant la méthode des plus proches voisins (Neighbor-Joining method : (Saitou et Nei, 1987)), et

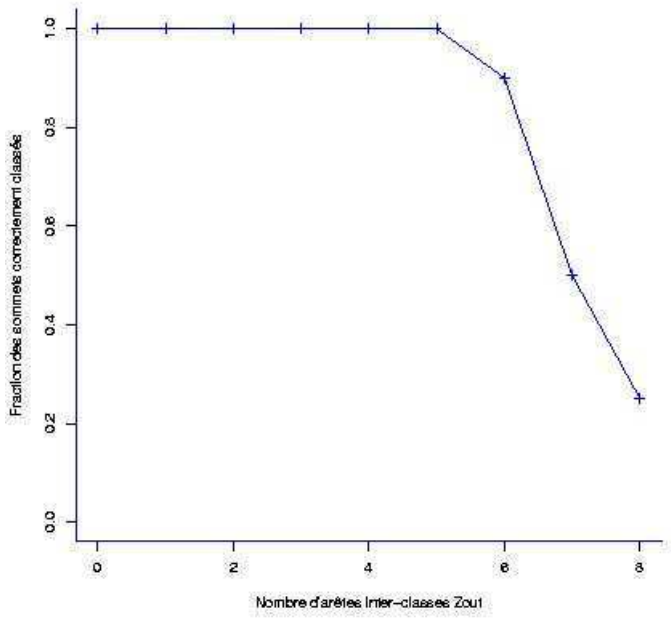


FIG. 3.29 – Résultats obtenus par Girvan et Newman sur 200 graphes aléatoires de 128 sommets répartis en 4 classes (d'après Girvan et Newman (2002)).

– par une méthode de partitionnement basée sur la recherche de similitudes (distance PAM) pour les séquences MSD et SBP (les séquences ne présentent pas assez de conservations pour être alignées : on recherche alors leur signature dans la proximité chromosomique des NBD).

Cette classification met en évidence le fait que pour un même transporteur ABC, les sous-familles de domaines sont compatibles. Ainsi, si dans un même système un domaine NBD de sous-famille A est associé à deux domaines MSD et SBP, alors tous deux appartiendront également à la sous-famille A. Ces résultats suggèrent que les différents partenaires des systèmes de transport ABC évoluent de façon concertée. Ils corroborent les résultats de Tomii et Kanehisa (1998) et de Saurin *et col.* (1994).

Dans un premier temps (chez *Bacillus subtilis*), douze sous-familles de transporteurs ABC ont été prédites. Par la suite, avec l'étude d'*Escherichia coli*, d'autres classes ont été ajoutées. Puis, avec l'accroissement du nombre de génomes entièrement séquencés, le nombre de classes a, peu à peu, augmenté. Ce sont ces classes que l'on cherche à raffiner ou tout au moins à retrouver.

3.3.2 Résultats

Nous avons ensuite appliqué notre algorithme aux transporteurs ABC de la famille A_5 impliquée dans le transport des oligosaccharides. Pour cela, nous nous sommes intéressés plus particulièrement aux protéines affines (SBP). Cette famille en comporte 765 dans les 95 génomes étudiés. La classification initiale (Quentin *et col.*, 1999) a permis d'identifier six classes.

Lors de cette étude nous avons relâché quelque peu les contraintes originales en considérant trois relations de similarité différentes entre les gènes :

– le meilleur score (BH),
– le meilleur score réciproque (BBH),
– et l'isorthologie.

Le choix de la relation de similarité dépend de la nature du problème à résoudre et permet des niveaux d'analyse différents. Idéalement, l'isorthologie devrait permettre de définir des groupes de transporteurs partageant la même spécificité de substrat, alors que BH et BBH devraient regrouper des systèmes partageant des substrats similaires.

Le nombre de composantes connexes (et donc le nombre minimal de classes) varie en fonction de la relation choisies. Nous avons une seule composante connexe avec la relation BH, onze composantes connexes[46] en utilisant BBH, et enfin quarante composantes connexes[47] avec l'isorthologie. Il faut noter que les classifications restent cohérentes : elles s'emboîtent presque parfaitement entre Isorthologie, BBH, et BH. Nous présentons en détail les résultats obtenus avec le BH sur la famille 5, car ils peuvent être directement confrontés aux classifications publiées. Les différents tests que nous avons réalisés ont montré que la fonction De_5 donnait les résultats les plus cohérents[48] par rapport à ceux de la classification d'ABCdb[49]. Le graphe obtenu est

[46] Sept d'entre elles contiennent moins de cinq éléments.

[47] Vingt-neuf d'entre elles contiennent moins de cinq éléments.

[48] On entend par "résultats cohérents" qu'ils ne surdécoupent pas trop le problème. Les fonctions De_3 et De_4 donnent elles aussi de bons résultats, compatibles avec la classification d'ABCdb, mais produisent plus de classes.

[49] Cette classification est effectuée à l'aide d'une méthode de partitionnement basée sur des recherches de similitudes pour les domaines MSD et SBP. Les prédictions sont validées par un expert.

Classe	Couleur	Classe	Couleur
1	jaune	10	turquoise
2	vert	11	vert foncé
3	rouge	12	gris
4	bleu	13	noir
5	rose clair	14	marron
6	blanc cassé	15	vert clair
7	orange	16	orange clair
8	violet	17	rose
9	bleu foncé	18	bleu-gris

TAB. 3.12 – Correspondance entre classes et couleurs de la figure 3.30.

présenté en figure 3.30 et les résultats sont résumés sous la forme d'un tableau à double entrées croissant (table 3.13). On trouve : en colonne, les six sous-familles connues pour cette classe – déterminées par expérimentation – et une famille désignée par "Autre" et qui contient les éléments qui n'étaient pas classés comme des SBP de la famille 5 (par exemple des éléments de la classe S_{17} dont la séparation S_5/S_{17} est mal définie) ; en ligne, les classes construites par notre algorithme. La première sous-famille, notée S_5, est une classe d'indécision : la procédure d'annotation automatique étant conservative, s'il y a un doute sur la sous-classe de la SBP considérée, elle est classée en S_5. Cet exemple peut paraître simple mais a l'avantage de rester lisible. Le graphe basé sur les MSD de la famille 5 avec la relation BH est donné comme exemple de graphe plus complexe (figure 3.31).

Les membres de la famille S_5 sont ventilés dans plusieurs sous-familles dont trois renferment uniquement des éléments de cette sous-famille (classes 1, 11, et 15) en très faible nombre. Par contre, la classe 10 qui contient 45 éléments de la famille S_5 et 5 éléments classés "Autre" pourrait correspondre à une nouvelle sous-famille qui n'avait pas été identifiée jusqu'à présent. Les autres membres de la famille S_5 pourraient appartenir aux sous-familles S_5a et S_5c. Nous pouvons remarquer que le nombre de classes est directement lié au nombre de membres de chaque sous-famille.

Les résultats obtenus présentent une très bonne adéquation avec la classification initiale. Ils suggèrent fortement la possibilité de raffiner le découpage en sous-familles, en particulier pour le groupe de 45 séquences annotées S_5 et appartenant à la classe 10.

Je ne donnerai pas le détail des résultats obtenus en utilisant les relations BBH et d'isorthologie car, malgré un nombre de classes beaucoup plus important, les résultats restent compatibles avec l'ancienne classification et sont donc redondants avec les résultats obtenus pour la relation BH. Nous obtenons 62 classes avec la relation BBH et 71 avec la relation d'isorthologie. Ces classes, beaucoup plus redécoupées par rapport à la partition BH, contiennent des éléments appartenant à la même classe initiale dans 95% des cas.

Une autre façon de valider notre méthode de classification est de croiser les résultats obtenus pour les différents partenaires des transporteurs ABC de la famille 5. En effet, si nous admettons que les gènes codant les différents partenaires d'un système évoluent en parallèle, alors

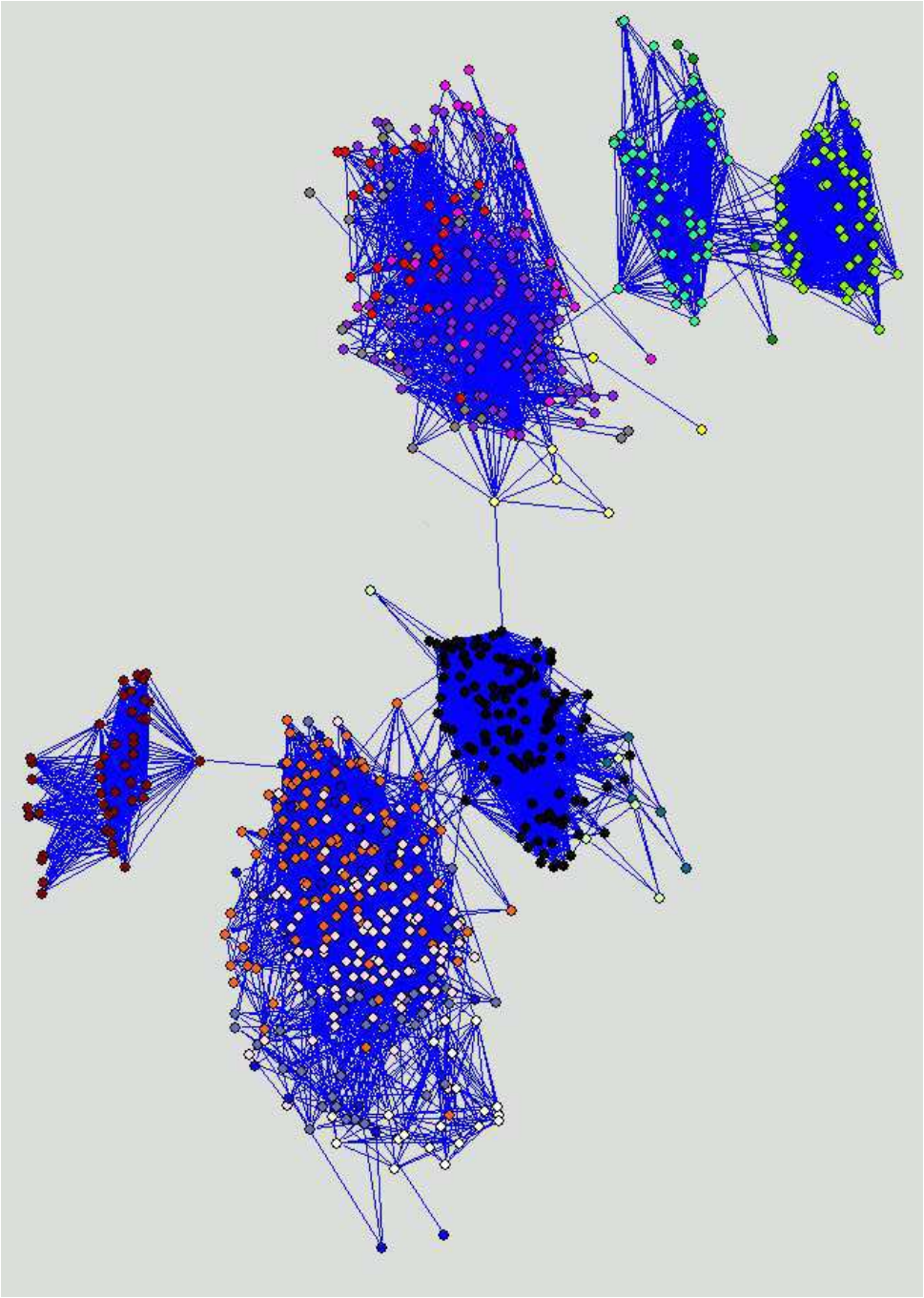


FIG. 3.30 – Graphe des SBP de la famille 5 des transporteurs ABC obtenu grâce au logiciel Pajek (Batagelj, 2001). Les gènes sont liés par des relations de similitude de type BH et les couleurs indiquent les classes détectées. La correspondance entre les classes et les couleurs est donnée en table 3.12.

	S_5	**S_5a**	**S_5b**	**S_5c**	**S_5d**	**S_5f**	**Autre**	
1	2	–	–	–	–	–	–	2
2	–	–	44	–	–	–	12	56
3	16	–	–	14	–	–	–	30
4	8	19	–	–	–	–	–	27
5	9	100	–	–	–	–	–	109
6	10	10	–	–	–	–	–	20
7	17	84	–	–	–	–	–	101
8	17	–	–	78	–	–	–	95
9	4	32	–	–	–	–	–	36
10	45	–	–	–	–	–	5	50
11	4	–	–	–	–	–	–	4
12	9	–	–	9	–	–	4	22
13	4	–	–	–	112	–	–	116
14	–	–	–	–	–	45	–	45
15	5	–	–	–	–	–	–	5
16	4	–	–	3	–	–	–	7
17	5	–	–	6	–	–	14	25
18	5	–	–	–	10	–	–	15
	164	245	44	110	122	45	35	765

TAB. 3.13 – Répartition des SBP des 6 sous-familles de la famille 5 des transporteurs ABC dans les classes calculées avec la relation BH et la fonction De_5.

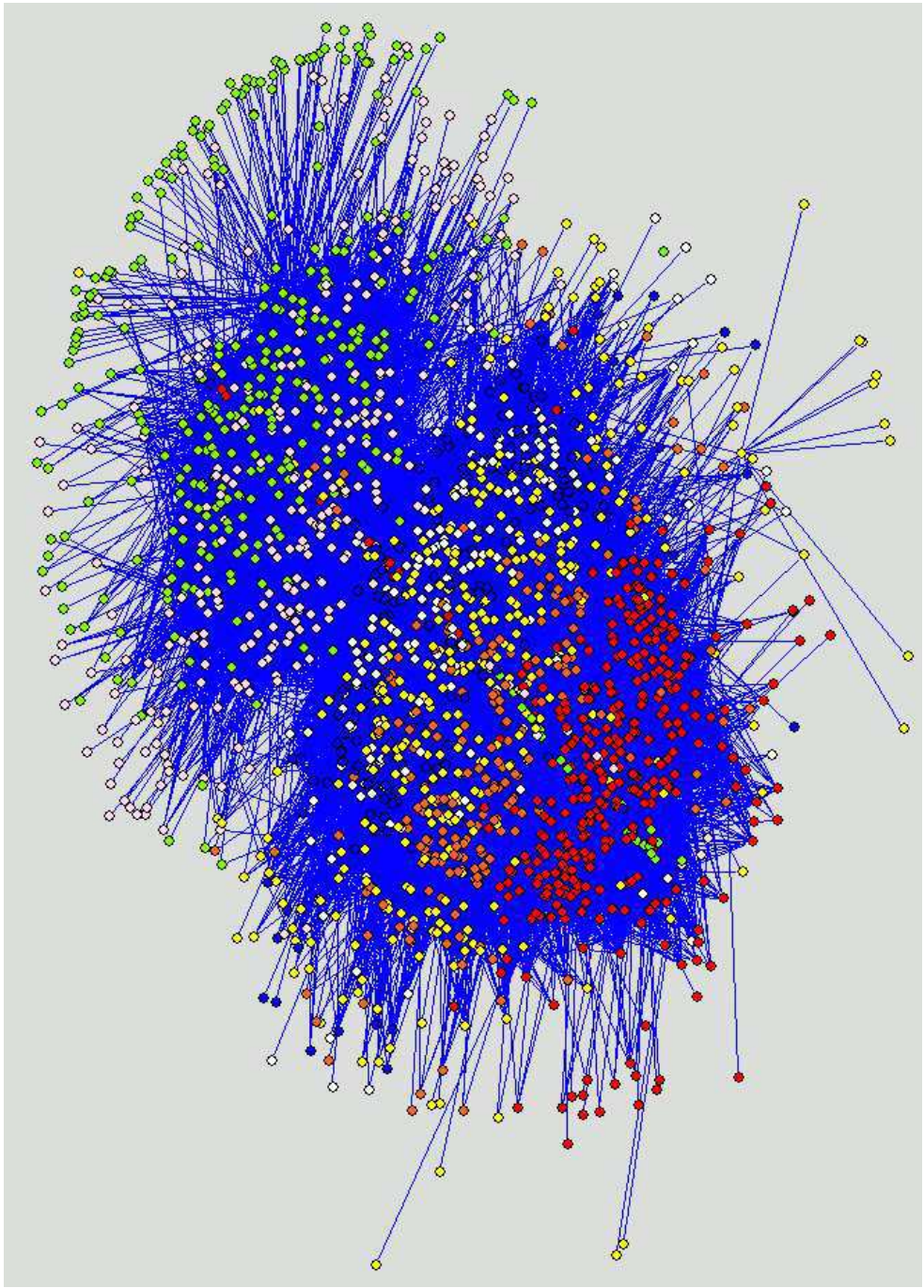


FIG. 3.31 – Graphe des MSD de la famille 5 des transporteurs ABC. Les gènes sont liés par des relations de similitude de type BH et les couleurs indiquent les classes détectées.

nous nous attendons à ce que les classifications obtenues sur chaque type de partenaire soient compatibles entre elles. En rapportant le nom de chaque gène au nom du système dans lequel il est impliqué, on peut effectuer le recoupement des trois classifications, et estimer la concordance des prédictions. Cette famille est composée de 1003 systèmes. La composition en domaines de ces systèmes est donnée en table 3.14. Sachant que pour avoir un système importeur complet il faut avoir au moins un domaine NBD, un domaine MSD et un domaine SBP, on ne considèrera que 505 systèmes, soit 50.3%, retenus pour l'expérience. Des résultats sur les NBD on peut dégager 8 classes et 7 classes des résultats sur les MSD[50]. Le recoupement des trois classifications est présenté en figure 3.32 (avec le détail des intersections entre éléments SBP/NBD et SBP/MSD en tables 3.15 et 3.16 où seul le meilleur score est conservé pour chaque ligne) où l'on peut s'apercevoir qu'il y a une concordance parfaite pour 433 systèmes, soit 85.7% des systèmes complets. Nous pouvons noter que le plus grand nombre de classes observées pour les SBP est compatible avec leur rôle dans le système de transport : elles donnent la spécificité de substrat aux transporteurs. Ainsi, la méthode de classification employée permet d'obtenir des résultats intéressants, validés d'un point de vue biologique et qui, de plus, vérifient les annotations de la base ABCdb tout en permettant d'améliorer, de raffiner la classification.

3.4 Conclusion & Perspectives

L'approche présentée, de par sa faible complexité, permet de traiter de grands graphes tels que les graphes de données biologiques. Les résultats ne tiennent compte que d'un seul domaine des systèmes de transport ABC. Pour toutes les familles, il faudrait effectuer les partitionnements sur tous les domaines puis recouper les informations, ce qui donnerait beaucoup plus de valeur aux prédictions formulées. On pourrait également tenir compte des relations de proximité entre domaines de manière à consolider les regroupements de transporteurs obtenus.
L'utilisation de relations de similarités différentes permet de faire varier le niveau d'analyse et d'obtenir des types d'information différents pour un même jeu de données.
Cette méthode peut également servir d'outil d'analyse des résultats de recherche de voisinage (cf Chapitre 2) : elle permettrait de détecter les groupes de gènes les mieux conservés.
Enfin, cette méthode pourrait permettre d'obtenir une nouvelle classification générale des transporteurs ABC, et pourrait même être appliquée à d'autres familles multigéniques.

[50]C'est le domaine SBP qui donne sa spécificité au système d'import. Il est donc normal d'obtenir moins de classes à partir d'une classification sur les autres domaines.

NBD	MSD	SBP	Nb. de systèmes
0	0	1	56
0	0	2	2
0	0	4	1
0	1	0	4
0	2	0	9
0	2	1	162
0	2	2	3
0	2	3	1
0	3	1	3
0	1	1	8
0	1	2	1
1	0	0	40
1	0	1	4
1	2	0	44
1	1	0	55
2	0	1	1
1	1	1	169
1	2	1	349
1	3	1	23
1	2	2	31
1	2	3	1
1	1	3	2
1	3	2	2
1	1	2	9
1	4	1	1
2	1	1	5
2	1	2	1
2	2	1	13
2	2	2	2
2	3	2	1

Tab. 3.14 – Composition en domaines des 1003 systèmes de la famille 5. Une ligne indique le nombre de domaines NBD, MSD et SBP des systèmes puis le nombre de systèmes ayant cette configuration. Les 498 systèmes de la première partie du tableau ne sont pas complets car il leur manque au moins un domaine. Les 505 systèmes de la seconde partie du tableau sont complets.

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
S_1			2					
S_2						45		
S_3			26					
S_4		22						
S_5		41						
S_6		1						
S_7		50						
S_8			75					
S_9		20						
S_{10}					39			
S_{11}					3			
S_{12}			19					
S_{13}	67							
S_{14}	16							
S_{15}								5
S_{16}			6					
S_{17}			19					
S_{18}		3						

Tab. 3.15 – Intersections entre éléments des classification SBP et NBD. Seuls les meilleurs scores sont conservés.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7
S_1		2					
S_2					54		
S_3		27					
S_4						19	
S_5				41			
S_6						1	
S_7						45	
S_8		75					
S_9				17			
S_{10}					38		
S_{11}					3		
S_{12}		18					
S_{13}							99
S_{14}							25
S_{15}							5
S_{16}		6					
S_{17}		22					
S_{18}							4

TAB. 3.16 – Intersections entre éléments des classification SBP et MSD. Seuls les meilleurs scores sont conservés.

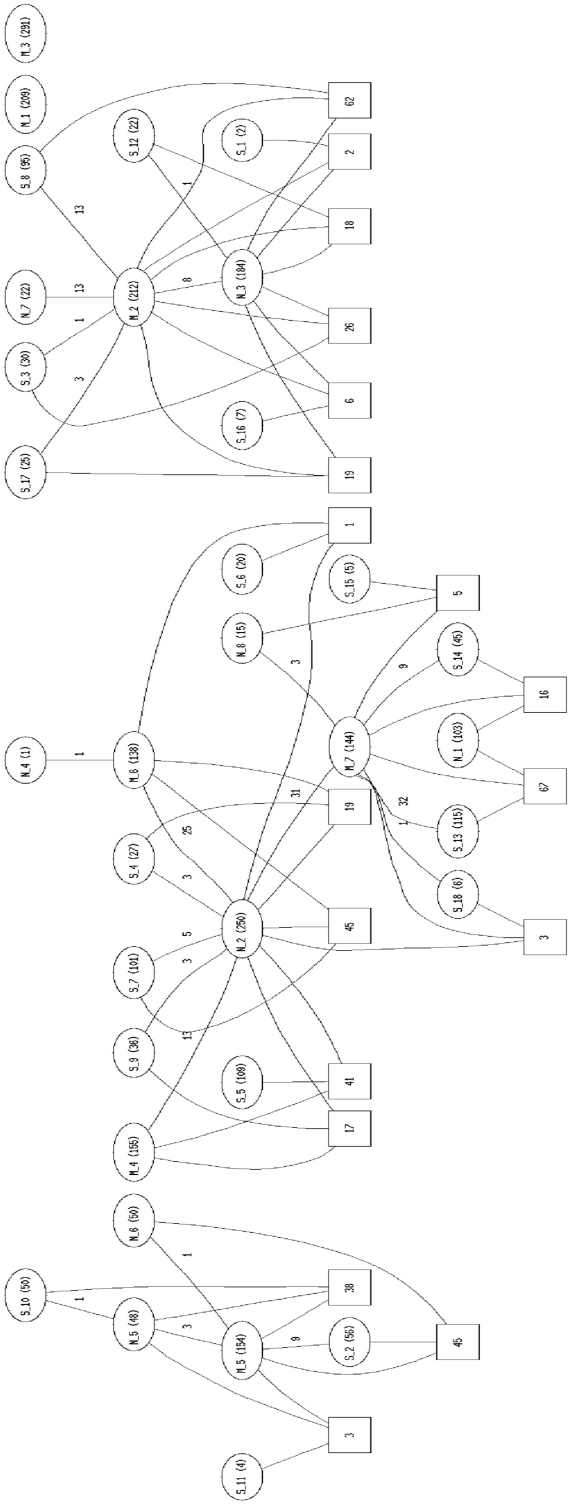


FIG. 3.32 – Recoupement des classifications basées sur les NBD, MSD et SBP. Les sommets sont étiquetés par le nom de la classe et le nombre d'éléments de cette classe : par exemple, N_x indique la classe x basée sur le graphe des NBD. Les arêtes valuées indiquent le nombre d'éléments communs entre deux classifications. Le nombre d'éléments communs à trois classification est indiqué à l'intérieur de carrés liés par des arêtes aux classes concernées (seuls les éléments les plus significatifs ont été représentés ici). La somme de ces éléments permet de voir que 433 éléments sont en accords à l'intérieur des trois classifications.

4

Reconstruction de systèmes incomplets

Un Anneau pour les amener tous et dans les ténèbres les lier.
J. R. R. Tolkien

Les transporteurs ABC sont des systèmes participant à l'export ou l'import de molécules diverses à travers la membrane. Ils sont composés d'une combinaison de trois types de domaines (NBD, MSD et SBP), généralement codés par des gènes différents (cf. Chapitre 1). Ils sont présents dans tous les génomes étudiés jusqu'à présent mais avec un nombre d'occurrences variable dépendant de la taille du génome (chez les bactéries ils représentent 5 à 6% des génomes[51]). Lors des étapes d'identification et de reconstruction des transporteurs ABC, la difficulté tient moins dans le nombre de partenaires que dans la faible conservation de séquence de certains d'entre eux. Ainsi, leur identification nécessite l'élaboration de stratégies complexes (Quentin *et col.*, 2002). Seuls les domaines NBD possèdent des séquences suffisamment conservées pour que l'on puisse les identifier sans ambiguïté (essentiellement par la reconnaissance des motifs impliqués dans la fixation et l'hydrolise de l'ATP). Les deux autres partenaires, MSD et SBP, ayant des séquences beaucoup moins conservées, il faut dans un premier temps identifier les NBD puis s'en servir de point d'ancrage pour rechercher dans leur voisinage des protéines ayant les caractéristiques des MSD et des SBP. Cette approche permet dans un premier temps de détecter les systèmes complets dont tous les domaines sont voisins sur le chromosome ((Fichant *et col.*, 1999) et (Quentin *et col.*, 2002)). Si cette méthode permet la reconstruction de la majorité des systèmes, elle ne permet pas de résoudre les cas où les gènes sont dispersés sur le chromosome (*systèmes éclatés*) et les cas où il manque un (souvent NBD) ou plusieurs (NBD et MSD) domaines (*systèmes incomplets*). Les premiers sont fréquents dans les génomes profondément remaniés alors que les seconds seraient le résultat de la duplication d'un système complet (duplication partielle ou duplication suivie de délétions de gènes). Deux problèmes sont

[51] Par exemple, dans le cas de *Mycoplasma genitalium* qui comporte 484 gènes, 40 gènes (soit 8% du génome) codent pour les 13 transporteurs ABC de cette bactérie.

donc posés : (i) reconstruire les systèmes complets lorsque les gènes sont éclatés sur le chromosome, (ii) compléter les systèmes partiels. Dans un premier temps une méthode, basée sur une analyse phylogénétique de gènes (analyse d'arbres) a été développée (Quentin *et col.*, 2002). Je commencerai par la présenter pour ensuite décrire un algorithme basé sur une analyse de relations évolutives. Puis, j'exposerai les résultats obtenus lors d'une étape de validation sur des systèmes que l'on sait reconstruire ou sur des systèmes déjà complets.

4.1 Méthode de reconstruction des transporteurs ABC basée sur une analyse d'arbres

La méthode de Quentin *et col.* (1999), basée sur une analyse d'arbres, est illustrée par l'exemple des transporteurs de sidérophores chez *Bacillus subtilis* (figure 4.33). Cette sous-famille est constituée de :

- quatre systèmes complets :
  - Yvr, composé d'un domaine NBD YvrA, d'un domaine MSD YvrB, et d'un domaine SBP YvrC.
  - Ycl, composé d'un domaine NBD YclP, de deux domaines MSD YclN et YclO, et d'un domaine SBP YclQ.
  - Yfm, composé d'un domaine NBD YfmF, de deux domaines MSD YfmD et YfmE, et d'un domaine SBP YfmC.
  - Fhu, composé d'un domaine NBD FhuC, de deux domaines MSD FhuB et FhuG, et d'un domaine SBP FhuD.
- un domaine NBD solitaire YusV,
- deux systèmes sans domaine NBD :
  - Yfi, composé de deux domaines MSD YfhA et YfiZ, et d'un domaine SBP YfiY.
  - Feu, composé de deux domaines MSD FeuB et FeuC, et d'un domaine SBP FeuA.
- et de deux domaines SBP solitaires YhfQ et YxeB.

A partir des séquences des trois types de domaines des transporteurs ABC, on construit trois arbres calculés à l'aide de la méthode NJ (Saitou et Nei, 1987). Ces arbres, représentant des relations de paralogie, sont ensuite utilisés pour associer les domaines solitaires (YusV, YhfQ, et YxeB) à un système. Le déroulement de la méthode est décrit en légende de la figure 4.33.
Cette méthode, appliquée à différents systèmes, a permis de démontrer le bien fondé de leur hypothèse : les gènes codant pour les différents partenaires d'un système présentent une évolution parallèle. Néanmoins, sa complexité de mise en oeuvre sur de gros jeux de données nous a conduit à développer une méthode reposant sur la même hypothèse mais d'implémentation plus aisée. Cette méthode repose sur l'exploitation des relations d'isorthologie et de voisinage.

4.2 Reconstruction des transporteurs ABC par analyse de graphes de relations évolutives

Ici, contrairement à l'exploration de voisinage (cf. Chapitre 2) ou à la recherche de synténies bactériennes, les gènes peuvent être dispersés sur le chromosome et ne conservent pas de structure ordonnée. L'idée est d'explorer les relations évolutives pour retrouver des systèmes pour

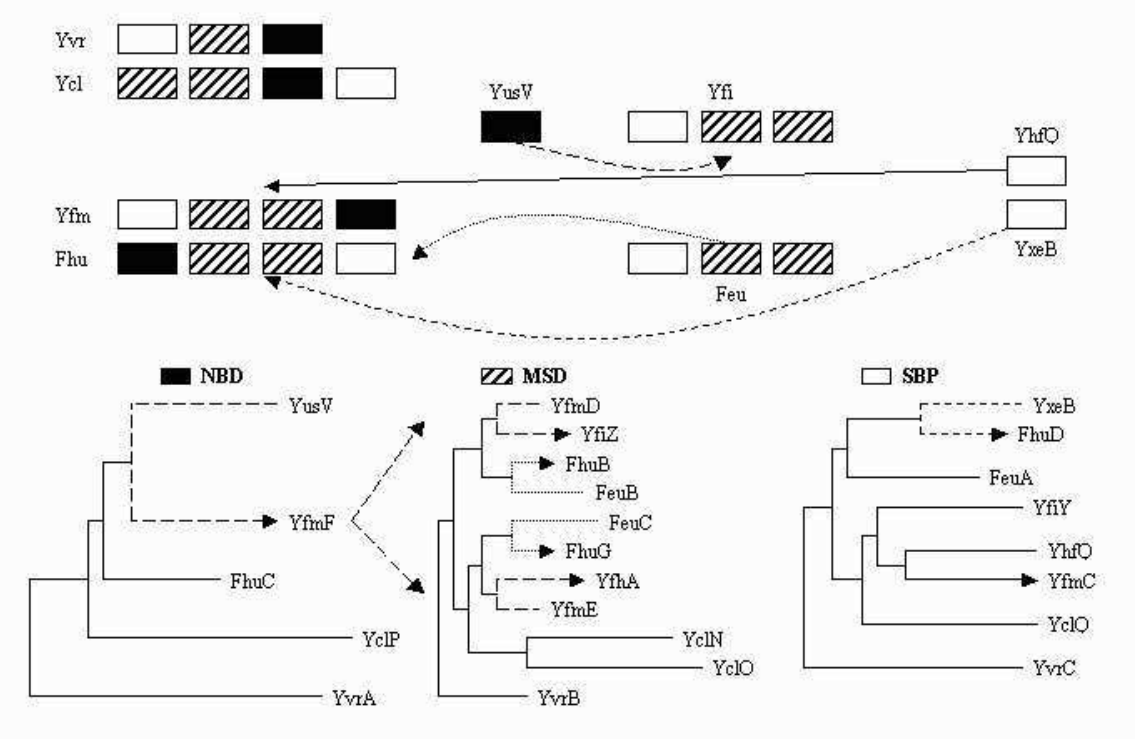


FIG. 4.33 – Reconstruction de systèmes incomplets de transporteurs de sidérophores chez *Bacillus subtilis* (d'après Quentin *et col.* (1999)). Sur l'arbre des SBP (an bas, à droite), YxeB est proche de FhuD et YhfQ est proche de YfmC. YxeB est prédit comme un gène codant pour un second domaine SBP du système Fhu (prédiction confirmant un résultat expérimental (Schneider et Hantke, 1993) a posteriori, alors que seule la protéine FhuD était connue). YhfQ est prédit comme un gène codant pour un second domaine SBP du système Yfm. Le même raisonnement peut être appliqué pour le gène codant pour un domaine NBD solitaire Yusv : sur l'arbre des NBD (en bas, à gauche), YusV est associée à YfmF, or les protéines membranaires, ou MSD, du transporteur Yfm sont proches de YfiZ et YfhA. YusV est prédit comme étant le domaine NBD énergisant le système Yfi. Pour le système Feu, comme il ne reste plus d'ATPase solitaire, son ATPase doit être recherchée parmi celles déjà attribuées à un autre système. Les protéines membranaires FeuB et FeuC sont proches de celles du système Fhu, et la protéine affine FeuA est proche de FhuD et YxeB. La prédiction est donc que le système Feu utilise le domaine NBD du transporteur Fhu pour énergiser son transport.

lesquels la proximité chromosomique est respectée. Nous utiliserons alors deux informations : la proximité physique des gènes et les relations évolutives (isorthologie). Le formalisme employé est inspiré de la méthode SNAP (pour **S**imilarity **N**eighborhood **Ap**proach – approche par similarité de voisinage) (Kolesov *et col.*, 2001). Cette méthode, partant d'un gène, permet de prédire les gènes qui lui sont fonctionnellement liés par utilisation de relations d'orthologie. Mais dans le cas des transporteurs ABC nous disposons d'informations supplémentaires avec la composition en domaines compatibles.
S'agissant d'une famille contenant de très nombreux paralogues, l'utilisation de l'orthologie au sens de Fitch (2000) n'est pas du tout envisageable. En effet, elle produirait un chaînage de gènes beaucoup trop important ; aucune information ne pouvant alors être extraite. Nous utiliserons donc la relation d'isorthologie, relation plus contraignante, générant moins de liens, mais permettant d'obtenir des résultats plus fiables.

Nous supposons ici que les étapes d'identification des protéines, et de classification en domaines ont été effectuées et que les données sont disponibles (nous ne travaillerons qu'avec les gènes identifiés comme partenaires d'un transporteur ABC). Nous connaissons alors les relations d'isorthologie entre gènes ainsi que les relations de proximité.

Notation 2.1 *Une relation d'isorthologie (cf Chapitre 1) entre deux gènes g_A et g_B (appartenant respectivement aux génomes A et B) est notée $I(g_A, g_B)$.*

Définition 2.2 *Une relation de voisinage entre deux gènes g_A et g'_A (appartenant à un même génome A) existe si g_A et g'_A sont consécutifs. Cette relation est notée $V(g_A, g'_A)$.*

Pour représenter simultanément les relations d'isorthologie et de voisinage, nous introduisons la notion de VI-graphe. Il s'agit d'un graphe où les sommets sont les gènes de transporteurs ABC, et où les arcs sont de deux types :
- les arcs I correspondant à la relation d'isorthologie qui est symétrique,
- et les arcs V correspondant à la relation de voisinage V qui est orientée (en fonction des brins).

Un exemple de problème et sa représentation sous forme de VI-graphe sont donnés en figure 4.34.
Notre objectif est ici de reconstruire des systèmes incomplets, partant d'un gène g_A du génome A, nous allons chercher à lui associer des gènes g^i_A du génome A. Pour former de telles paires (dans le cas où le système de transport n'est constitué que de deux gènes, il n'y aura qu'une seule paire) nous allons rechercher dans le VI-graphe un ensemble de chemins (voir figure 4.35) à partir desquels nous allons construire un graphe de paires de gènes candidats à la constitution d'un système. Ce graphe, appelé C_A-graphe, se définit par :

Définition 2.3 *Soit $\Gamma = (S, A, v)$ un graphe non orienté valué. Les sommets S sont les gènes, et les arcs A sont les paires de candidats. La fonction de valuation v représente la confiance que l'on peut accorder à la prédiction d'une paire candidate.*
(g_A, g'_A) est une arête du C_A-graphe Γ si et seulement s'il existe un génome B et une chaîne de longueur quelconque de gènes consécutifs (appartenant à un système de transport) dans B

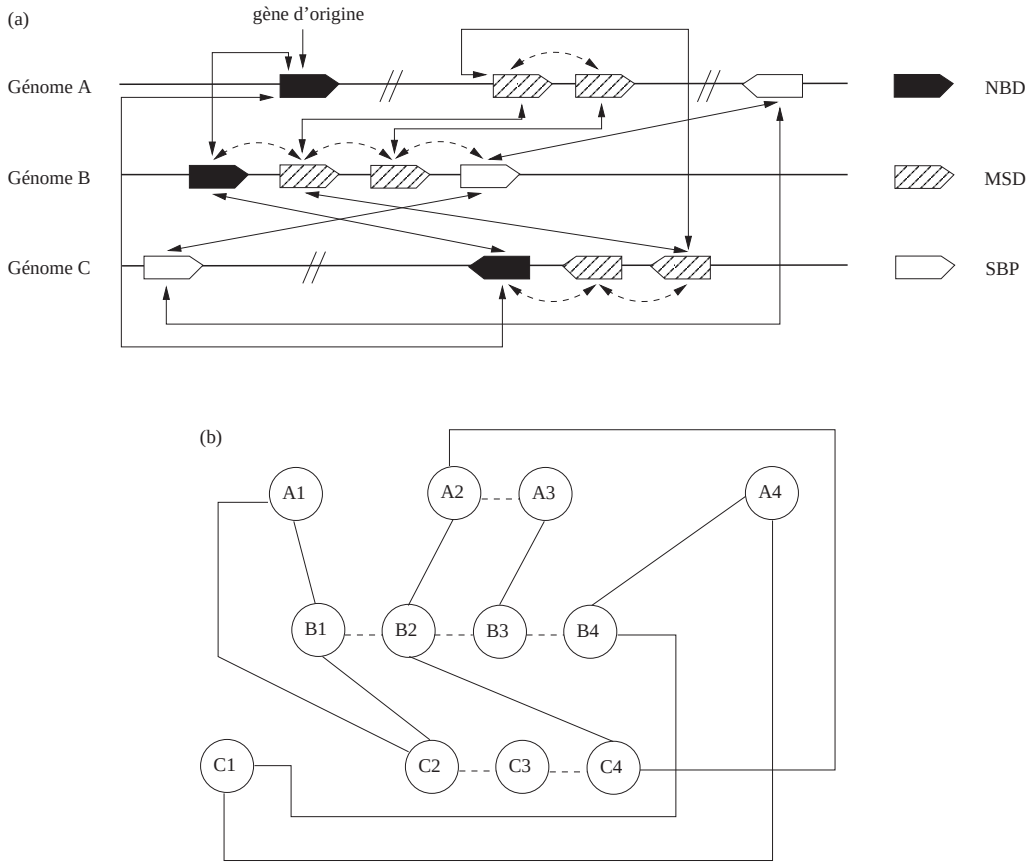


FIG. 4.34 – (a) Conservation des groupements de gènes sur 3 génomes A, B, et C, et (b) VI-graphe correspondant. Les flèches ou arêtes en pointillés indiquent une V-relation et les flèches ou arêtes pleines indiquent une I-relation. Les noms des gènes du VI-graphe sont formés à partir du nom du génome et de la position du gène en lisant de gauche à droite. Ainsi, $C3$ correspond au troisième gène du génome C.

$(g_B^1, ..., g_B^n)$ *au sens de la relation* V*, avec* $I(g_A, g_B^1)$ *et* $I(g_B^n, g'_A)$.

En effectuant cette recherche de candidats pour tous les partenaires de transporteurs ABC disponibles, nous obtenons une liste de paires de candidats. En reprenant l'exemple de la figure 4.34, si l'on veut calculer les paires candidates à partir du gène A_1, on obtient :

Données			Résultats
I-relation	V-relation(s)	I-relation	Paire candidate
$I(A_1, B_1)$	$V(B_1, B_2)$	$I(B_2, A_2)$	(A_1, A_2)
$I(A_1, B_1)$	$V(B_1, B_2), V(B_2, B_3)$	$I(B_3, A_3)$	(A_1, A_3)
$I(A_1, B_1)$	$V(B_1, B_2), V(B_2, B_3), V(B_3, B_4)$	$I(B_4, A_4)$	(A_1, A_4)
$I(A_1, C_2)$	$V(C_2, C_3), V(C_3, C_4)$	$I(C_4, A_2)$	(A_1, A_2)

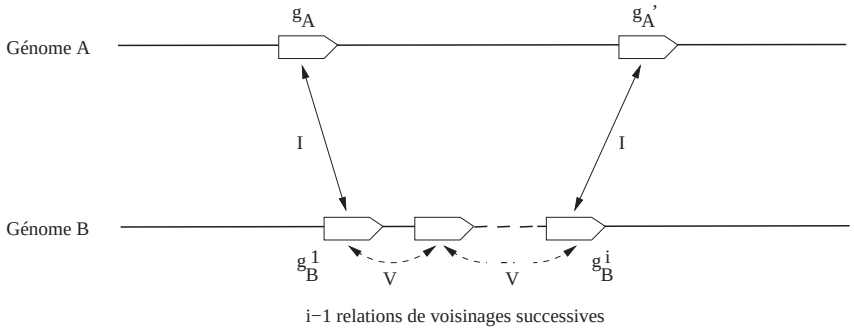


FIG. 4.35 – Détermination d'un couple de gènes candidat à la constitution d'un système par recherche d'un chemin particulier dans un VI-graphe. Les gènes g_A et g'_A ne sont pas nécessairement voisins.

Pour un même gène de départ nous obtenons de nombreux couples candidats différents qui ne seront pas forcément compatibles : si deux systèmes ont divergé (paralogie) et ont peu évolué, il est possible d'obtenir des paires de candidats pointant du premier système vers le second. Une paire (NBD système 1, NBD système 2) pourrait alors représenter un "bruit" et ne serait pas informative pour reconstruire le système. C'est pourquoi nous avons introduit un indice de confiance : il s'agit d'une pondération du nombre d'apparition de chaque couple candidat (g, g') par le nombre total de couples candidats ayant pour gène d'origine g.

Définition 2.4 *Le poids d'une paire candidate est donné par $w(g, g')$: nombre d'occurences de la paire (g, g').*

Définition 2.5 *L'indice de confiance d'un couple candidat (g, g') est :*

$$conf(g, g') = \frac{w(g, g')}{\sum_i w(g, g_i)}$$

Ainsi, les candidats de la table précédente ont une confiance de :

Couple candidat	Indice de confiance
(A_1, A_2)	$2/4 = 0.5$
(A_1, A_3)	$1/4 = 0.25$
(A_1, A_4)	$1/4 = 0.25$

Nous pouvons maintenant définir la fonction de valuation du C_A-graphe. Il s'agit de la confiance que l'on peut accorder à la prédiction d'une paire de candidat :

Définition 2.6 *La fonction de valuation d'un C_A-graphe $\Gamma = (S, A, v)$ est donnée par : $v(x, y) = min(conf(x, y), conf(y, x))$.*

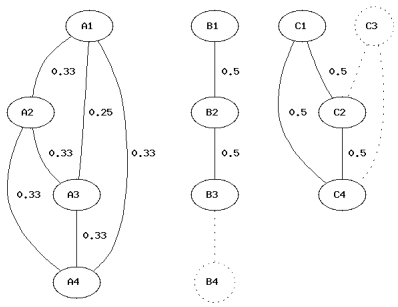


FIG. 4.36 – Graphe des gènes issus de la figure 4.34 et candidats à la constitution d'un système. Les gènes et les arcs indiqués en pointillés ont été ajoutés d'après les relations de proximité.

Nous avons décidé de fixer un seuil de confiance (après expérimentation de diverses valeurs) : tout couple dont l'indice de confiance est inférieur à 0.1 sera rejeté. On déconnecte alors dans le C_A-graphe toutes les arêtes (x, y) telles que $v(x, y) < 0.1$. Les composantes connexes indiquent les prédictions de systèmes. Sur l'exemple très simple de la figure 4.34, nous obtenons le graphe de la figure 4.36. On peut y voir que les systèmes prédits sont : $\{A_1, A_2, A_3, A_4\}$, $\{B_1, B_2, B_3, B_4\}$ et $\{C_1, C_2, C_3, C_4\}$.

La qualité des prédictions obtenues dépend directement du nombre de relations d'isorthologie associées au système étudié et donc du nombre de génomes pris en compte. Contrairement aux prédictions de fonctions réalisées dans le chapitre 2, ce sont les génomes les moins éloignés (d'un point de vue taxonomique) du génome d'intérêt qui sont susceptibles de fournir le plus d'informations. En effet, avec l'augmentation des distances évolutives, la probabilité de rencontrer des paralogues sur les trajectoires évolutives augmente et la fréquence des relations d'isorthologie diminue.

4.3 Résultats

Dans un premier temps, nous avons repris l'exemple de la reconstruction des transporteurs de sidérophores décrite en figure 4.33.

4.3.1 Application à un cas connu : les transporteurs de sidérophores

La méthode précédente (Quentin *et col.*, 1999) exploitait uniquement des données de paralogie (analyse de la famille chez *Bacillus subtilis*). Ici, nous utilisons uniquement les relations d'isorthologie calculées sur une cinquantaine de génomes. Autre point important, dans un premier temps, l'information sur le voisinage des gènes n'a pas été utilisée pour le système à reconstruire : on choisit le gène de départ et on recherche ses partenaires sans connaître ses voisins. Ce dispositif permet d'évaluer la méthode par rapport aux résultats de la figure 4.33 qui exploitait ces relations de voisinage et la paralogie. Les résultats sont présentés en figure 4.37 et résumés

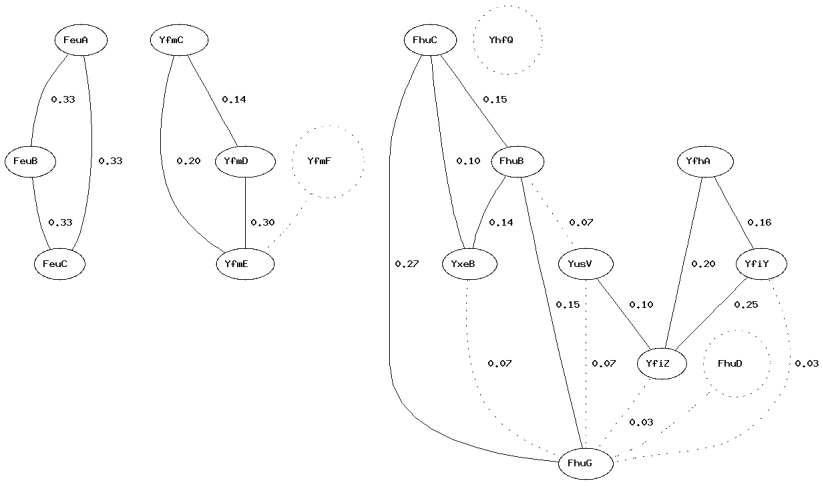


FIG. 4.37 – Graphe non orienté et valué des candidats à la constitution d'un système. Les arcs en pointillés sont les arcs supprimés car le seuil de confiance était inférieur à l'indice de confiance minimal 0.1. Les sommets indiqués en pointillés ont été ajoutés d'après les relations de proximité.

en table 4.17.
Si les relations de voisinage ne sont pas prises en compte, quatre systèmes sont reconstruits : Feu, Fhu, Yfi et Yfm. Les trois gènes fhuD, yfmF, et yhfQ ne possèdent pas les relations d'isorthologie nécessaires à l'élaboration d'un chemin (figure 4.37) et ne peuvent donc pas être assemblés. A ce niveau, l'utilisation des relations de paralogie pourrait permettre de compléter les systèmes. Nous avons choisi de ne pas utiliser cette information car, pour un gène dans un génome, il n'existe au plus qu'un plus proche paralogue. On ne devrait alors se fier qu'à une unique information lors de l'assemblage : on préfère ne pas "sur-prédire". Par contre, l'utilisation de la proximité chromosomique (ou voisinage) permet d'améliorer la prédiction : nous pouvons prédire que FhuD sera lié au système Fhu et que YfmF sera lié au système Yfm.
Comme on peut le voir, par notre méthode, le nombre d'erreurs (nombre de systèmes incorrectement reconstruits) est nul et, seulement trois gènes ne possèdent pas de relation d'isorthologie permettant de déterminer un chemin. En y ajoutant les informations sur la proximité, un système n'est pas totalement reconstitué par rapport aux résultats de la méthode d'arbres (FeuC/FhuC) et la protéine YhfQ reste solitaire. Etudions maintenant globalement la qualité des résultats obtenus.

4.3.2 Etude plus générale

Afin d'évaluer la méthode à une plus grande échelle, nous avons construit un test à partir des systèmes complets contenus dans la base de données. Pour cela, nous avons traité les partenaires des systèmes complets comme des protéines isolées et nous avons utilisé notre méthode pour retrouver leurs partenaires dans le système fonctionnel. La comparaison des prédictions avec les données de la base permet d'évaluer l'efficacité de la méthode au travers du taux de systèmes

Partenaire	Type	Voisinage	Prédiction	Préd.+Voisin.	Méthode d'arbres
FeuA	SBP	FeuC	FeuC	FeuC	FhuC
FeuB	MSD	FeuC	FeuC	FeuC	FhuC
FeuC	MSD	FeuC	FeuC	FeuC	FhuC
FhuB	MSD	FhuC	FhuC	FhuC	FhuC
FhuC	NBD	FhuC	FhuC	FhuC	FhuC
FhuD	SBP	FhuC	NC	FhuC	FhuC
FhuG	MSD	FhuC	FhuC	FhuC	FhuC
YxeB	SBP	YxeB	FhuC	FhuC	FhuC
YfhA	MSD	YfiY	YfiY	YfiY	YfiY
YfiY	SBP	YfiY	YfiY	YfiY	YfiY
YfiZ	MSD	YfiY	YfiY	YfiY	YfiY
YusV	NBD	YusV	YfiY	YfiY	YfiY
YfmC	SBP	YfmF	YfmF	YfmF	YfmF
YfmD	MSD	YfmF	YfmF	YfmF	YfmF
YfmE	MSD	YfmF	YfmF	YfmF	YfmF
YfmF	NBD	YfmF	NC	YfmF	YfmF
YhfQ	SBP	YhfQ	NC	YhfQ	YfmF

Tab. 4.17 – Résultats obtenus par proximité, prédiction, prédiction et proximité, méthode d'arbres de Quentin *et col.* (1999) sur l'exemple de la figure 4.33. Une mention NC indique une absence d'information et donc une prédiction impossible.

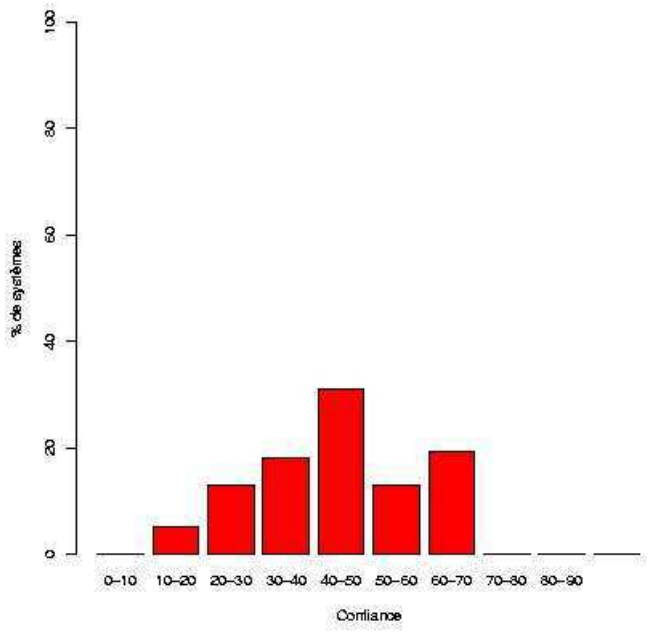


FIG. 4.38 – Etude de l'indice de confiance des systèmes correctement reconstruits.

correctement assemblés et de la valeur de l'indice de confiance. A ce niveau, nous préférons faire moins de bonnes prédictions que de risquer d'en faire des mauvaises.
Les résultats ont été obtenus sur une centaine de génomes (5632 systèmes de transport) de la base ABCdb. Le taux d'erreur est très faible (à peu près 0,05% des systèmes sont mal assemblés) et ne correspond peut être pas réellement à des erreurs mais à des systèmes qui auraient été remaniés. Par contre, l'indétermination avoisine les 60%. Ce phénomène, comme nous l'avons vu précédemment, s'explique par le fait qu'il existe des cas où l'on ne peut déterminer les isorthologues (cf figure 4.40 ci-après), et d'autres où l'indice de confiance en chacun des liens est inférieur au seuil minimum. Le taux de systèmes correctement prédit avoisine quant à lui les 40% sans prise en compte des données sur le voisinage du système de départ. Sinon, dans ce cas trivial, les prédictions correctes approcheraient bien entendu les 100%.

J'ai ensuite étudié la valeur moyenne de l'indice de confiance obtenue globalement pour chacun des systèmes correctement reconstruits (moyenne des indices des partenaires de chaque système). Les résultats sont présentés en figure 4.38. Comme nous l'avons vu précédemment, l'indice de confiance dépend du nombre de partenaires constituant le système : les plus fortes valeurs d'indice correspondent donc, soit à de petits systèmes correctement reconstruits, soit à des systèmes ayant peu évolué et dont les isorthologues sont très proches dans chacun des génomes. Pour la moitié des systèmes reconstruits, l'indice de confiance se situe entre 30% et 50%, ce qui montre que les paires conservées permettent de constituer des systèmes relativement stables.

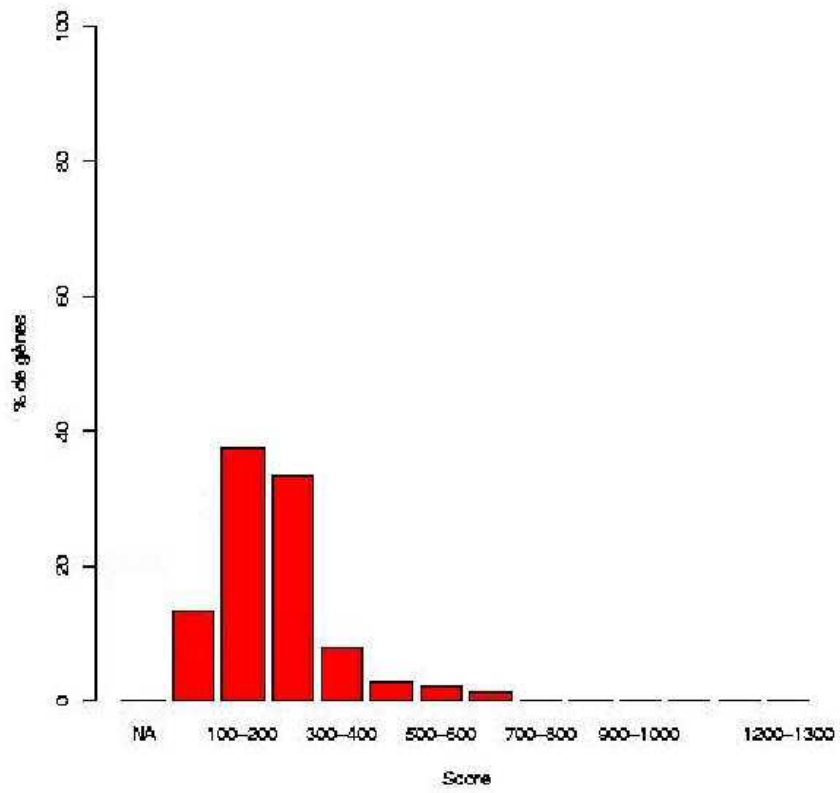


FIG. 4.39 – Etude du score d'isorthologie des gènes des systèmes correctement reconstruits.

Pour finir, j'ai étudié le score d'isorthologie des gènes des systèmes correctement reconstruits (figure 4.39) puis des systèmes indéterminés ou mal reconstruits (figure 4.40). Le score est ici exprimé par plage de 100 et 'NA' indique l'absence de lien d'isorthologie. Nous avons ici la preuve que la reconstruction échoue principalement en l'absence de relations d'isorthologie. Dans les cas de reconstruction correcte, les scores sont bien sûr plus élevés.

4.4 Conclusion & Perspectives

Cette méthode automatique de reconstruction des systèmes incomplets est donc globalement satisfaisante. Elle devrait être intégrée en tant que méthode dans la base de connaissances ABCkb (Capponi *et col.*, 2001). Pour l'instant, seule une interface web en CGI-Perl permet d'en faire un outil d'aide à la décision pour la reconstruction des systèmes dans la base ABCdb. Nous avons vu que le point faible de cette méthode apparaissait lors de l'absence de relations d'isorthologie clairement définies. Il faudrait tester la qualité des prédictions en combinant les informations

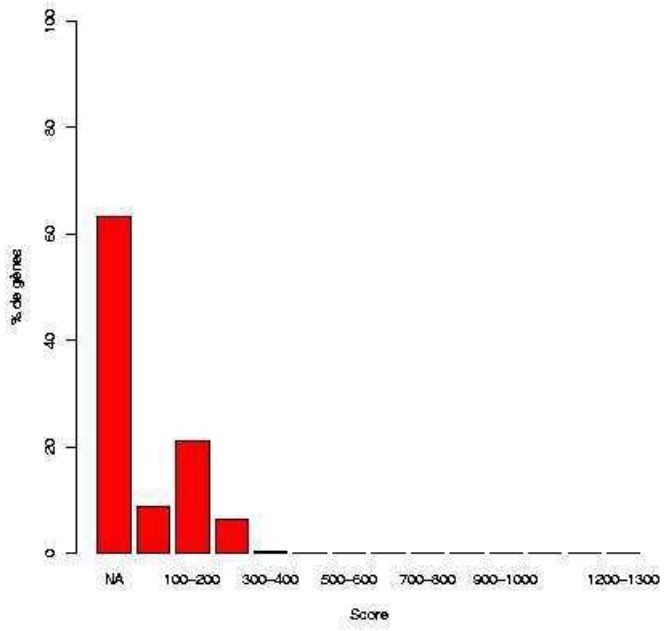


FIG. 4.40 – Etude du score d'isorthologie des gènes des systèmes indéterminés ou mal assemblés.

d'isorthologie et le meilleur score d'homologie réciproque : en augmentant la taille des données, le nombre de cas indécis devrait chuter.
La notion de proximité sur le chromosome peut être étendue dans une sorte de base d'apprentissage des systèmes étudiés : seule la notion de voisinage est alors modifiée mais il existe toujours une relation entre gènes permettant de prédire un couple de candidats à la construction d'un système.
L'utilisation d'une autre relation que l'isorthologie est tout à fait envisageable pour des familles de gènes où les paralogues sont moins fréquents. D'autre part, nous avons déterminé l'indice de confiance minimum de façon arbitraire. Il faudrait essayer de déterminer cet indice en fonction des données et non plus comme un paramètre de la méthode.
Enfin, cette méthode pourrait être généralisée pour être appliquée à d'autres problèmes biologiques tels que les voies métaboliques. En effet, possédant des informations dans quelques génomes, il serait alors possible de détecter les liens fonctionnels entre des gènes d'autres génomes.

Conclusion

Les travaux présentés dans cette thèse trouvent leur source dans un problème ou une hypothèse biologique. Ils ont tous pour objectif d'aider à l'analyse des transporteurs ABC dans les génomes bactériens. L'une de nos préoccupations principales est la complexité en temps des méthodes développées : devant la masse de données à traiter, qui est en constante augmentation, les algorithmes doivent être nécessairement rapides.
Des méthodes de résolution ont été développées pour répondre aux problèmes : (i) d'identification du substrat transporté, (ii) de classification des transporteurs par classes de substrat, et (iii) de reconstitution des systèmes incomplets. Elles ont toutes permis de vérifier la validité des hypothèses employées : (i) des gènes voisins sur le chromosome peuvent être impliqués dans un même processus métabolique s'ils ont été conservés au cours de l'évolution, et (ii) des gènes présentant des similarités de séquence peuvent permettre la synthèse de protéines de même fonction. Les résultats obtenus en appliquant ces méthodes aux données biologiques sont encourageants : sur des jeux de données restreints ils ont permis de valider les méthodes, mais leur mise en oeuvre sur l'ensemble des données relatives aux transporteurs ABC n'a pas encore été effectuée. Résumons maintenant l'ensemble de ce travail, du point de vue des objectifs atteints et des perspectives qu'il permet d'envisager.

Identification du substrat
La recherche de synténies bactériennes est beaucoup trop contraignante du fait de la notion d'ordre qu'elle impose au niveau des ensembles de gènes conservés entre espèces. La suppression de l'ordre permet, chez les transporteurs ABC, d'obtenir de meilleurs résultats. La méthode présentée dans cette thèse est basée sur l'ancrage sur un gène dont on explore le voisinage. Cette exploration est réglementée par des contraintes et ne porte, pour l'instant, que sur des comparaisons entre deux espèces ; la comparaison multiple peut être effectuée par combinaison de comparaisons deux à deux. L'évolution prochaine de cette méthode se décompose en deux étapes :

- au niveau algorithmique : il faut étendre les contraintes sur de multiples génomes sans perte d'information, c'est-à-dire qu'un groupe de gènes conservé dans une fraction de la totalité des génomes étudiés doit pouvoir être détecté. De plus, les gènes devront cette fois être traités par domaines.
- au niveau interface utilisateur : il faut rendre cette interface beaucoup plus souple en autorisant le choix d'une relation d'homologie (orthologie, orthologie au sens de COG, isorthologie, ...). Les données utilisées devront être accessibles de manière plus simple, dans une base de données relationnelle. La présentation actuelle des résultats n'est pas satisfaisante : il faut présenter des statistiques sur les gènes et les groupes conservés et pouvoir "désactiver" certains gènes de la recherche pour pouvoir tester leur influence.

Les objectifs sont ici d'améliorer l'algorithme et de produire un outil d'analyse des données

totalement achevé et réellement utilisable. Pour des raisons de souplesse, d'ergonomie et de compatibilité, l'emploi du langage C pour la partie algorithmique et de PHP/MySQL pour la partie mise en oeuvre est privilégié.
Le traitement automatique des résultats pourrait être effectué par recherche de densité (cf Chapitre 3). Ceci permettrait de retrouver les gènes dont les liens fonctionnels sont les plus significatifs.

Classification des transporteurs ABC
Pour chaque domaine des transporteurs ABC, en représentant les relations de similitude par un graphe, la recherche des zones de forte densité permet de déterminer des classes de substrat transporté. Ces classes dépendent fortement de la relation employée : au plus cette relation est contraignante, au plus le nombre de classes est important et précis. En ramenant le nom de chaque protéine à celui du transporteur ABC auquel elle appartient il est alors possible d'étudier les intersections de classes entre les partitionnements obtenus pour les différents domaines. Ces intersections de classes représentent les éléments pour lesquels il n'y a aucune ambiguïté possible. La méthode de recherche de zones denses n'a pas encore été utilisée massivement sur toutes les données de transporteurs ABC. Voici quelques directions d'étude qui pourraient se révéler intéressantes :

- La recherche de meilleures fonctions de densité est toujours d'actualité. Grâce à la généralisation de ces fonctions d'après une distance, on peut s'inspirer de distances locales.
- L'intégration de la modélisation particulière des transporteurs ABC au coeur de l'algorithme devrait conduire à des résultats plus précis. Les intersections de classes sont effectuées après les partitionnements sur les différents domaines. La prise en compte de ces informations dès le départ permettrait d'être plus proche du problème. Ainsi, la prise en compte des relations de proximité par exemple, permettrait de traiter les transporteurs ABC non plus par domaines, mais en tant qu'entité.

Une interface de ce programme en CGI-Perl a été créée mais la poursuite de ce travail passe obligatoirement par le développement d'un outil de liaison entre les prédictions effectuées et la base de données ABCdb de manière à rendre les données obtenues accessibles et utilisables.
La recherche de classes dans les graphes peut avoir de multiples applications en dehors des transporteurs ABC, notamment avec les réseaux métaboliques ou encore, un domaine extérieur à la biologie : la fouille de données textuelles.

Reconstitution des systèmes incomplets
Nous pouvons reconstituer des systèmes pour lesquels des gènes codant pour certains partenaires sont absents du regroupement ou que tout ou partie des gènes est dispersé sur le chromosome. Cette méthode fonctionne très bien sur l'exemple restreint donné en chapitre 4 et sur la majorité des systèmes que l'on peut reconstruire grâce à la proximité chromosomique. Toutefois, des améliorations ou de nouveaux développements sont à envisager :

- Le seuil de confiance est déterminé de façon arbitraire. Il doit exister un moyen de l'obtenir en fonction des données du graphe.
- La prise en compte des domaines des protéines dans le calcul des relations d'isorthologie doit permettre une amélioration des prédictions.
- Cette méthode croît en efficacité avec le nombre de génomes dont nous disposons. Or les

données d'isothologie doivent être calculées et le processus est très long. Il faudrait être en mesure de travailler sur le maximum de génomes possible.

Ici encore, les développements logiciels ne sont qu'expérimentaux et les prédictions ne sont pas directement liées à la base de données ABCdb.

La méthode d'identification du substrat transporté pourrait très rapidement donner naissance à un logiciel de l'envergure de STRING. Pour les autres méthodes, les développements à faire sont un peu plus nombreux avant de pouvoir créer des modules incorporables à la base de données ABCdb ou à la base de connaissances ABCkb.

Ces méthodes sont liées par un même objectif : recueillir des informations sur les transporteurs ABC et permettre leur annotation automatique. Différentes approches permettent de collecter des informations diverses et complémentaires. C'est pourquoi je ne me suis pas attaché à un problème particulier, mais plutôt à son environnement, aux problèmes qui se rattachaient au domaine des transporteurs ABC. La diversité des méthodes ne doit pas masquer l'unicité d'hypothèses biologiques et particulièrement l'utilisation des relations évolutives. Ces relations sont à la base de chacune des méthodes présentées : c'est à partir de ces dernières que les prédictions sont formulées ; il faut donc apporter une grande rigueur à leur acquisition et améliorer leur détection.

Les trois méthodes pourraient être mises en commun pour former un outil de détection de réseaux métaboliques ou certains éléments pourraient se retrouver éloignés sur le chromosome alors que d'autres seraient restés dans la même proximité. Donc bien qu'éloignées méthodologiquement et en apparence au niveau des applications, ces trois méthodes se rejoignent dans un même objectif : la découverte de réseaux fonctionnels au travers de relations évolutives.

Bibliographie

Allen, J. F. (1983) Maintaining knowledge about temporal intervals. *CACM*, **26**, 832–843.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. et Lipman, D. J. (1990) Basic local alignement search tool. *J. Mol. Biol.*, pp. 403–410.

Altschul, S. F., Madden, T. L., Schäfer, A. A., Zhang, J. et Zhang, Z. (1997) Gapped BLAST and PSI-BLAST : A new generation of protein database searh programs. *Nucl. Acids Res.*, **25**, 3389–3402.

Bader, G. D. et Hogue, C. W. (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, **4**.

Barabási, E. A.-L. et Albert, R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

Barriel, V. et Tassy, P. (2003) Analyser les caractères et reconstruire la phylogénie du monde vivant. *Dossier SagaScience Evolution - CNRS (http ://www.cnrs.fr).*

Batagelj, V. (2001) Pajek - program for large networks analysis and visualization. In *Link Analysis and Visualization.*

Bergeron, A., Corteel, S. et Raffinot, M. (2003) The algorithmic of gene teams. In Guigo, R. et Gusfield, D. (eds.), *Workshop on Algorithms in Bioinformatics*, pp. 464–476. LNCS.

Birge, E. A. (1994) *Bacterial and Bacteriophage genetics.* 3rd edition.

Bussey, H., Storms, R. K., Ahmed, A., Albermann, K. et et col., E. A. (1997) The nucleotide sequence of *Saccharomyces cerevisiae* chromosome XVI. *Nature*, pp. 103–105.

Capponi, C., Chabalier, J., Quentin, Y. et Fichant, G. (2001) A knowledge base for biological integrated systems. *IEEE Intelligent Systems, Special Issue : Intelligent Systems in Biology*, **16**, 52–60.

Chandon, J. L. et Pinson, S. (1981) *Analyse Typologique - Théories et applications.* Masson.

Christie, P. J., Korman, R. Z., Zahler, S. A., Adsit, J. C. et Dunny, G. M. (1987) Two conjugation systems associated with *Streptococcus faecalis* plasmid pcf10 : identification of a conjugative transposon that transfers between *S. faecalis* and *Bacillus subtilis. J. Bacteriol.*, **169**, 2529–2536.

Claverie, J.-M., Audic, S. et Abergel, C. (2000) La bioinformatique : une discipline stratégique pour l'analyse et la valorisation des génomes. `http://igs-server.cnrs-mrs.fr`.

Colombo, T. (2004) Biograph : a perl library to manipulate graphs of biological data. `http://www.cpan.org/~baldr`.

Colombo, T., Benhamou, B., Capponi, C., Fichant, G. et Quentin, Y. (2001) Inférence fonctionnelle par l'analyse du contexte génétique - une application aux transporteurs ABC. In *Entretiens J. Cartier on Comparative Genomes et Poster JOBIM (2002)*. Lyon. Communication orale.

Colombo, T., Guénoche, A. et Quentin, Y. (2002) Inférence fonctionnelle par l'analyse du contexte génétique - une application aux transporteurs ABC. In *Journées ALBIO*. Montpellier. Communication orale.

Colombo, T., Guénoche, A. et Quentin, Y. (2003) Recherche de quasi-cliques d'orthologues pour la prédiction fonctionnelle. In Sanjuan, E. (ed.), *Proc. JIM*, pp. 203–212. Metz.

Colombo, T., Guénoche, A. et Quentin, Y. (2004) Looking for high density areas in a graph - application to orthologous genes. *submitted to Discrete Applied Mathematics*.

Danchin, A. (1998) *La barque de Delphes*. Odile Jacob.

Dandekar, T., Snel, B., Huynen, M. et Bork, P. (1998) Conservation of gene order : a fingerprint of proteins that physically interact. *Trends Biochem Sci.*, **23**, 324–328.

Darlu, P. et Tassy, P. (1993) *Reconstruction Phylogénétique - Concepts et méthodes*. Masson edition.

Dassa, E., Hoffnung, M., Paulsen, I. T. et Saier, M. H. (1993) The *Escherichia coli* ABC transporters : an update. *Mol. Microbiol.*, **32**, 887–889.

Dassa, E., Hofnung, M., Paulsen, I. T. et Saier, M. H. (1999) The *Escherichia coli* ABC transporters : an update. *Mol. Microbiol.*, **32**, 887–889.

Dechter, R., Meiri, I. et Pearl, J. (1991) Temporal constraint satisfaction problems. *Artificial Intelligence*, **49**, 61–95.

Decottignies, A. et Goffeau, A. (1997) Complete inventory of the yeast ABC proteins. *Nat. Genet.*, **15**, 137–145.

Diday, E. (1971) Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *Revue de Statistiques appliquées*, **19**.

Douarin, N. L. (2000) *Des chimères, des clones et des gènes*. Odile Jacob.

Enault, F., Suhre, K., Poirot, O., Abergel, C. et Claverie, J. (2004) Phydbac 2 : improved inference of gene function using interactive phylogenomic profiling and chromosomal location analysis. *Nucl. Ac. Res.* Accepted.

Etienne, J. (1999) *Biochimie génétique - Biologie moléculaire*. Masson, 5ème edition.

Fath, M. J. et Kolter, R. (1993) ABC transporters : bacterial exporters. *Microbial. Rev.*, **57**, 995–1017.

Felsenstein, J. (1989) Phylip : Phylogeny inference package. *Cladistics*, **5**, 164–166.

Fichant, G., Quentin, Y. et Denizot, F. (1999) Analyse du répertoire des transporteurs de type ABC chez bacillus subtilis. *Mésogée*, **57**, 41–46.

Fitch, W. M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–113.

Fitch, W. M. (2000) Homology : a personal view on some of the problems. *Trends in Genetics*, **16**.

Fleishmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R. et col. (1995) Whole genome random sequencing and assembly of *Haemophilus unfluenzae Rd*. *Science*, **269**, 496–512.

Gaspin, C., Bessière, C., Moisan, A. et Schiex, T. (1995) Satisfaction de contraintes et biologie moléculaire. *Revue d'Intelligence Artificielle*.

Girvan, M. et Newman, M. E. J. (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, **99**, 7821–7826.

Glover, F. et Laguna, M. (1997) Tabu search. *Kluwer Academic Publishers*. Dordrecht.

Guyader, H. L. (2003) L'évolution biologique dans les théories et dans les faits. *Dossier Saga-Science Evolution - CNRS (http ://www.cnrs.fr)*.

Guénoche, A. (2003) Partitions optimisées selon différents critères : évaluation et comparaison. *Mathématiques et Sciences Humaines*, **161**, 41–58.

Guénoche, A. (2004) Clustering by graph density. In *International Federation of Classification Societies*. Chicago.

Guénoche, A., Hansen, P. et Jaumard, B. (1991) Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *J. Classification*, pp. 5–30.

Hansen, P. et Jaumard, B. (1997) Cluster analysis and mathematical programming. *Mathematical Programming*, **79**, 191–215.

Henikoff, S. et Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.

Higgins, C. F. (1992) ABC transporters : from micro organism to man. *Annu. Rev. Cell Biol.*, **8**, 67–113.

Hinton, J. C. D. (1997) The *Escherichia coli* genome sequence : the end of an era or the start of the fun ? *Mol. Microbiol.*, **26**, 417–422.

Holland, B. et Blight, M. (1999) ABC-ATPases, adaptable energy generators fuelling transmembrane movement of a variety of molecules in organisms from bacteria to humans. *J. Mol. Biol.*, **293**, 381–399.

Holland, J. H. (1975) *Adaptation in natural and artificial systems.* Univ. of Mishigan Press.

Huynen, M. A. et Bork (1998) Measuring genome evolution. In *Proc. Natl. Acad. Sci. USA*, volume 95, pp. 5849–5856.

Huynen, M. A. et Snel, B. (2000) Gene and context : integrative approaches to genome analysis. *Adv Protein Chem.*, **54**, 345–379.

Ihaka, R. et Gentleman, R. (1996) R : A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Jacob, F. et Monod, J. (1961) On the regulation of gene activity. *Cold Spring Harbor Symposia on Quantitative Biology*, **126**, 193–211.

Jacquard, A. (1992) *La légende de la vie.* Flammarion.

Jacquard, A. et Kahn, A. (2001) *L'avenir n'est pas écrit.* Bayard.

Jain, A. K., Murty, M. N. et Flynn, P. J. (1999) Data clustering : a review. *ACM Computing Surveys.*

Janvier, P. (2003) Les caractères, mémoire de l'évolution. *Dossier SagaScience Evolution - CNRS (http ://www.cnrs.fr).*

Joseph, P., Fichant, G., Quentin, Y. et Denizot, F. (2002) Genetic link between ABC permease and regulatory systems in the bacillus/clostridium group suggests an involvement in a common physiological process. *J. Mol. Microbiol. Biotechnol.*, **4**, 503–513.

Karlin, S. et Altschul, S. F. (1993) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Nat. Acad. Sci. USA*, **87**, 2264–2268.

Kirkpatrick, S., Gellat, C. D. et Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, pp. 671–680.

Kolesov, G., Mewes, H.-W. et Frishman, D. (2001) Snapping up functionnaly related genes based on context information : a colinearity-free approach. *J. Mol. Biol.*, **311**, 639–656.

Lathe, W. C., Snel, B. et Bork, P. (2000) Gene context conservation of a higher order than operons. *TIBS*, **25**, 474–479.

Lawrence, J. G. (1999) Gene transfer, speciation, and the evolution of bacterial genomes. *Curr Opin Microbiol.*, **2**, 519–523.

Luc, N., Risler, J.-L., Bergeron, A. et Raffinot, M. (2003) Gene teams : a new formalization of gene clusters for comparative genomics. *Comput. Biol. and Chem.*, **27**, 59–67.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings 5th Berkeley Symposium*, **1**, 281–297.

Marcotte, E. M., Pellegrini, M., Thomson, M. J., Yeates, T. O. et Eisenberg, D. (1999) A combined algorithm for genme-wide prediction of protein function. *Nature*, **402**, 83–86.

Matic, I. (1995) Les mécanismes du contrôle des échanges génétiques interspécifiques et de la variabilité génétique chez les bactéries. *Bull. Inst. Pasteur*, pp. 187–219.

Matsuda, H., Ishihara, T. et Hashimoto, A. (1999) Classifying molecular sequences using a linkage graph with their pairwise similarities. *Theoretical Computer Science*, **210**, 305–325.

Mazumder, R., Kolaskar, A. et Seto, D. (2001) Geneorder : comparing the order of genes in small genomes. *Bioinformatics*, **17**, 162–166.

Milo, R., Kashtan, N., Itzkovitz, S., Newman, M. E. J. et Alon, U. (2004) On the uniform generation of random graphs with prescribed degree sequences. *submitted to Phys. Rev. E.*

Montanari, U. (1974) Networks of constraints : fundamental properties and application to picture processing. *Information Sciences*, **7**, 95–132.

Morgat, A. et Viari, A. (2001) Synténies bactériennes. In *Entretiens J. Cartier on Comparative Genomes*. Lyon. Communication orale.

Mushegian, A. R. et Koonin, E. V. (1996) Gene order is not conserved in bacterial evolution. *Trends Genet.*, **12**, 289–290.

Médigue, C., Bocs, S., Labarre, L., Mathé, L. et Vallenet, D. (2002) L'annotation in silico des séquences génomiques. *Médecine/Sciences*, **18**, 237–250.

Newman, M. E. J. (2001) Scientific collaboration networks : Shortest paths, weighted networks, and centrality. *Phys. Rev. E*, **64**.

Newman, M. E. J. (2003) *Random graphs as models of networks.* Wiley - VCH.

Newman, M. E. J. (2004) Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, **in press**.

Newman, M. E. J., Strogatz, S. H. et Watts, D. J. (2001) Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, **64**.

Nicolas, F. (2003) *Analyse comparative et évolutive des répertoires de transporteurs* ABC *dans les génomes bactériens séquencés.* Mémoire de DEA, Université d'Aix-Marseille II.

Nitschke, P., Guerdoux-Jamet, P., Chiapello, H., Faroux, G. et Henaut, C. (1998) Indigo : a world wide web review of genomes and genes functions. *FEMS Microbiol. Rev.*, **22**, 207–227.

Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D. et Maltsev, N. (1999) The use of gene clusters to infer functionnal coupling. In *Proc. Nat. Acad. Sci. USA*, volume 96, pp. 2896–2901.

Pasek, S., Bergeron, A., Risler, J.-L., Louis, A., Ollivier, E. et Raffinot, M. (2004) Identification of genomic features using domain teams. *Proceedings of JOBIM.*

Paulsen, I. T., Sliwinski, M. K. et Saier, M. H. J. (1998) Microbial genome analyses : global comparisons of transport capabilities based on phylogenies, bioenergetics and substrate specificities. *J. Mol. Biol.*, **277**, 573–92.

Perrière, G., Duret, L. et Gouy, M. (2000) HOBACGEN : database system for comparative genomics in bacteria. *Genome Res.*, **10**, 379 – 385.

Prim, R. C. (1957) Shortest connection networks and some generalizations. *Bell System Technical Journal*, **36**, 1389–1401.

Quentin, Y., Chabalier, J. et Fichant, G. (2002) Strategies for the identification, the assembly and the classification of integrated biological systems in completely sequenced genomes. *Computers and Chemistry*, **26**, 447–457.

Quentin, Y. et Fichant, G. (2000) ABCdb : an ABC transporter database. Assembly and analysis of ABC trasnport systems in complete genomes. *J. Mol. Microbiol. Biotechnol.*, **2**, 501–504.

Quentin, Y., Fichant, G. et Denizot, F. (1999) Inventory, assembly and analysis of *Bacillus subtilis* ABC transport systems. *J. Mol. Biol.*, **287**, 467–484.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. et Parisi, D. (2003) Defining and identifying communities in networks. Preprint cond-mat/0309488.

Rand, W. M. (1971) Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**, 846 – 850.

Rensberger, B. (2000) *Au coeur de la vie.* De Boeck Université.

Rocha, E. P. C. (2000) *Analyse exploratoire des génomes bactériens.* Ph.D. thesis, Université de Versailles Saint-Quentin-en-Yvelines.

Rougemont, J. et Hingamp, P. (2003) DNA microarray data and contextual analysis of correlation graphs. *BMC Bioinformatics*, **4**.

Saitou et Nei (1987) Neigbor joining method. *Mol. Biol. Evol.*, **4**, 406–425.

Saurin, W., Koster, W. et Dassa, E. (1994) Bacterial binding protein-dependant permeases : characterization of distinctive signatures for functionnaly related integral cytoplasmic membrane proteins. *Mol. Microbiol.*, **12**, 993–1004.

Schneider, S. et Hantke, K. (1993) Iron hydroxamate uptake systems in *Bacillus subtilis* : identification of a lipoprotein as part of a binding protein-dependent transport system. *Mol. Microbiol.*, **8**, 111–121.

Snel, B., Lehmann, G., Bork, P. et Huynen, M. A. (2000) STRING : a web-server to retrieve and display the repeatedly occuring neighbourhood of a gene. *Nucleic Ac. Res.*, **28**, 3442–3444.

Suhre, K. et Claverie, J. (2004) FusionDB : a database for in-depth analysis of prokaryotic gene fusion events. *Nucl. Ac. Res.*, **32**, 273–276.

Suyama, M. et Bork, P. (2001) Evolution of prokaryotic gene order : genome rearrangements in closely related species. *Trends in Genetics*, **17**, 10–13.

Taglicht, D. et Michaelis, S. (1998) *A complete catalogue of Saccharomyces cerevisiae ABC proteins and their relevance to human health and disease*. Academic Press.

Tam, R. et Saer, M. H. (1993) Structural, functional, and evolutionary relationships among extracellular solute-binding receptors of bacteria. *Microbiol. Rev.*, **57**, 320–346.

Tamames, J., Casari, G., Ouzounis, C. et Valencia, A. (1997) Conserved clusters of functionally related genes in two bacterial genomes. *J. Mol. Evol.*, **44**, 66–73.

Tamames, J., Gonzalez-Moreno, M., Mingorance, J., Valencia, A. et Vicente, M. (2001) Bringing gene order into bacterial shape. *Trends in Genetics*, **17**, 124–126.

Tatusov, R. L., Koonin, E. V. et Lipman, D. J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S., Kiryutin, B., Galperin, M. Y., Fedorova, N. D. et Koonin, E. (2001) The COG database : new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Ac. Res.*, **29**, 22–28.

Tomii, K. et Kanehisa, M. (1998) A comparative analysis of ABC transporters in complete microbial genomes. *Genome Res.*, **8**, 1048–1059.

Trémolières, R. et Vanbaelinghem, M. (1977) La méthode de percolation pour l'analyse des données. *Etudes et Documents série Recherche*. IAE d'AIx-en-Provence.

Trémolières, R. C. (1994) *Percolation and multimodal data structuring*, volume 263 - 268 of *New Approaches in Classification and Data Analysis*. Springer-Verlag.

von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P. et Snel, B. (2003) STRING : a database of predicted functional associations between proteins. *Nucleic Acids Res.*, **31**, 258–261.

Watson, J. D. et Crick, F. A. C. (1953) Molecular structure of nucleic acids. A structure for desoxyribose nucleic acid. *Nature*, **171**, 737–738.

Watts, D. J. et Strogatz, S. H. (1998) Collective dynamics of 'small-world' networks. *Nature*, **393**, 440–442.

Whitman, W. B., Coleman, D. C. et Wiebe, W. J. (1998) Prokaryotes : the unseen majority. *Proc. Natl. Acad. Sci. USA*, **95**, 6578 – 6583.

Wishart, D. (1969) *Mode analysis : A generalization of nearest neighbour which reduces chaining effects*, pp. 282–311. Numerical Taxonomy. Academic Press. London.

Algorithmes pour la recherche de classes de gènes en relations fonctionnelles par analyse de proximités et de similarités de séquences

Résumé : Notre étude porte sur les transporteurs ABC dans les génomes bactériens complets. L'analyse bioinformatique du répertoire de ces systèmes comprend l'identification des partenaires, l'assemblage, la reconstruction des systèmes incomplets, la classification en sous-familles, et l'identification du substrat transporté. Cette thèse propose des outils permettant l'étude de ces problèmes par l'utilisation de méthodes informatiques. Les hypothèses biologiques employées sont que : (i) des gènes voisins sur le chromosome peuvent être impliqués dans un même processus métabolique s'ils ont été conservés au cours de l'évolution, et (ii) des gènes présentant des similarités de séquence peuvent permettre la synthèse de protéines de même fonction.

Trois études ont été menées sur le répertoire des transporteurs ABC :

- L'exploration du voisinage chromosomique. D'après l'hypothèse selon laquelle plus les gènes conservés dans le voisinage d'un transporteur sont proches, plus leur lien fonctionnel avec le transporteur est fort, on essaye d'identifier le substrat transporté ou des associations de gènes. Ce problème est traité par une méthode de résolution issue des problèmes de satisfaction de contraintes.
- La classification. Les transporteurs ABC sont classés par grandes catégories en fonction des molécules qu'ils transportent (sucres, ...). Pour chaque domaine, en représentant les relations d'homologie par un graphe, la recherche des zones de forte densité permet de déterminer des sous-classes de substrat.
- La reconstitution des systèmes incomplets. Les transporteurs ABC sont assemblés en utilisant la proximité chromosomique des gènes codant pour les domaines et la compatibilité des sous-familles de domaines. Lorsque la proximité n'est pas respectée, on utilise une stratégie développée à partir d'une méthode d'analyse de graphes pour assembler les domaines et prédire des systèmes actifs.

Ces méthodes, en complément de l'identification des partenaires et de l'assemblage, permettent une étude fonctionnelle des transporteurs ABC. Elles pourraient être appliquées à d'autres systèmes biologiques.

Mots-clés : bioinformatique, transporteur ABC, partitionnement de graphe, synténie, satisfaction de contraintes.

Algorithms for the research of functionnaly related classes of genes by the analysis of proximities and of the similarities of sequences

Abstract : Our study focuses on ABC transporters in complete bacterial genomes. The bioinformatic analysis of these systems includes the identification of partners, the assembly, the reconstruction of incomplete systems, the classification in sub-families, and the identification of the carried substrate. This thesis proposes tools allowing the study of these problems by using computational methods. The biological hypotheses employed are : (i) neighbor genes on the chromosome can be implicated in a same metabolic process if they are conserved during evolution, and (ii) genes with similarities of sequences can allow the synthesis of proteins of the same function.

Three studies have been made on ABC transporters :

- The exploration of chromosomical neighborhood. According to the hypothesis which says that the closer the genes conserved in the neighborhood of a transporter are, the stronger their functionnal link with the transporter is, we try to identify the carried substrate or associations between genes. This problem is treated by a resolution method stemming from the constraints satisfaction problems.
- Classification. ABC transporters are classified into big categories in function of the molecules they carry (sugars, ...). For each domain, by representing the homological relations by a graph, the research for the high density areas allow us to determine sub-classes of substrate.
- The reconstitution of incomplete systems. ABC transporters are assembled using the chromosomical proximity of the genes coding for the domains, and the compatibility of the sub-families of the domains. When the proximity is not respected, we use a strategy developped from a method of graph analysis to assemble the domains and predict the active systems.

These methods, complementary to the identification of partners and of the assembly process, allow a functional study of the ABC transporters. They could be applied to other biological systems.

Keywords : bioinformatics, ABC transporter, graph partitionning, syntenie, constraints satisfaction.

LCB – 31 chemin Joseph Aiguier – 13009 Marseille
IML – 163 avenue de Luminy– 13009 Marseille